MOTOROLA

SB9600 Serial Bus Protocol Manual

Version 05.13
August 28, 1995

Copyright © Motorola, Inc. 1991,1992,1994,1995, All Rights Reserved.

Revision History

Point of Contact:

Person(s): Jerry Sandvos
Organization: Radio Software Technology Center
Group: Radio Products Group
Sector: Land Mobile Products Sector
Mail Drop: FL08 Room 1183
Address: 8000 W. Sunrise Blvd, Ft. Lauderdale, FL 33322
Phone Number: (305) 723-3726
Email Address: EPOR67@email.mot.com

Revision History

Revision #	Date	Author	Description
	10/85		Table of Contents:Changed to include Appendix A. Revision History: Added to alert readers to the latest changes. Page 7: Change TXAUD and RXAUD sensitivity to 35ua and FLTAUD sensitivity to 87mv. Page 25: Misspelled ”transmitter” in 14th line of text. Page 31: Add TXINH opcode for trunking-type options. Page 33: Add TXINH opcode for trunking-type options. Page 35: Add TXINH opcode for trunking-type options. Page 47: Alert tone opcode expanded to include priorities, 100 Hz increments, and more patterns. Page 52: RF frequencies of Motorola trunking example corrected. Page 64: The system will Reset after modifying EEPROM. Page 87: TXAUD operation modified to more correctly handle the situation where options of different bands use the line simultaneously. Page 88.1: Add TXINH opcode for trunking-type options. Appendix A: Lists the alert tones available with ALRTTN.

Revision #	Date	Author	Description
	12/85		Page 9: Change ”low” Reset to an ”active” Reset. Page 15: Change ”low” Busy to an ”active” Busy. Page 17: Change ”low” Busy to an ”active” Busy. Page 22: Add ”one” to 9th line from the bottom. Page 70: Add bit to PTTINH opcode to distinguish between two types of inhibits (radio-only or system-wide). Page 90: Modify reference made to invalid transmit frequency. Appendix A: Correct Talk Permit Tone description.

Revision #	Date	Author	Description
	06/86		Table of Contents: Changed to include Appendix B and Appendix C. Page 7: Correct and clarify TXAUD and RXAUD sensitivity (again). Page 15: Clarification: Change ”return” to ”transmit” in 2nd line. Page 24: Ignore overrun/framing errors. Page 26: Clarify what transmitting device does upon collision detection. Page 27: A device responding to a request must respond to NAKs. Page 31: Add SGINFO Request, ACDADJ, OSCVAL, and PALIM. Page 33: Add SGINFO Request, ACDADJ, OSCVAL, and PALIM. Page 34: Add SGINFO Request, ACDADJ, OSCVAL, and PALIM. Page 37.2: Add ACDADJ opcode to allow option deviation control. Page 46: Change ACPRVL so it operates independently of RADKEY. Page 55: Change DEVVAL so it is restored on Active Mode changes, or system reset. Extend full-scale value. Page 57: Extend DISMUT to mute options which have discriminator audio for their input. Page 67.1: Add OSCVAL opcode to program oscillator frequency. Page 67.2: Add PALIM opcode to adjust TX power limit circuitry. Page 76: Radio will send RXPLIN when the internal PL decoder changes state due to HUB or Monitor button operation. Page 82: SGINFO more thoroughly defined. Appendix B: List other pertinent SYS9000 documents for reference. Appendix C: List option characteristics.

Revision #	Date	Author	Description
	12/86		Page 7 : Make signal levels dependent on allowable peak deviation. Page 15: Add NAK timing to Request length calculation. Page 30: PRTVAL also has the ADDRESS of the destination device. Page 31: Add PRTVAL, TRKOPC, TRKMDU, ALARMS, PWRCTL, ACTSTU Page 31: Change TXINH to TSTMOD Page 33: Add PRTVAL, TRKOPC, TRKMDU, ALARMS, PWRCTL, ACTSTU Page 33: Change TXINH to TSTMOD Page 34: Add PRTVAL, TRKOPC, TRKMDU, ALARMS, PWRCTL, ACTSTU Page 34: Change TXINH to TSTMOD Page 38: ACTMDU indicates a conventional mode only. Page 46.1: Add ACTSTU to inform options of misc radio states. Page 46.2: Add ALARMS to inform options which alarms are active. Page 62: Reference ACTSTU Page 68.1: Add PRTVAL to read port contents. Page 68.2: Add PWRCTL to control current drain and sleep modes. Page 72: Add DATA1 and DATA2 to RADRDY opcode. Page 77: Reference ACTSTU Page 79: Reference ACTSTU Page 85: Reference ACTSTU Page 86.1: Add TRKMDU to inform options of a trunked active mode. Page 86.2: Add TRKOPC for custom trunking opcodes. Page 86.3: Add TSTMOD to put devices into a predetermined test. Page 88: TXCTRL counter also cleared on active mode changes. Page 88.1: Delete TXINH Appendix A: Add Volume Set tone Appendix C: Add Single Tone and Vehicular Repeater option.

Revision #	Date	Author	Description
	06/87		Page 31: Add RXMODE request. Page 32: Add TXMODE request. Page 33: Add RXMODE request and TXMODE request. Page 34: Add RXMODE request and TXMODE request. Page 47: To turn off continuous tone, DATA4 must be 0, not DATA3. Page 68: Add Audio Detect bit to PLDECT. Page 68.2: Add Display dim capability to PWRCTL. Page 83: Add Audio Detect bit to SQLDET. Appendix A: Add Extended Talk Permit Tone. Appendix C: Correct DTMF priority.
Version 5.0	03/11/91		First release in Interleaf format. Page numbers will now differ from previous versions. All Sections:made document generic - not related to any specific portable or mobile radio. All Sections:Fixed Many spelling and typographical errors. Section 3.CHNUMB is changed to CHLNUM (was spelled both ways throughout document) Eliminated Framing Protocol (Fast Programming Protocol will be added in Version 5.1) SFTPT1 - 56:Opcode added as a generic way to tune radio.

Revision #	Date	Author	Description
Version 5.1	12/11/92		Added Appendix I to describe SBEP (Serial Bus Expanded Protocol). TSTMOD is now usable only by radio (address $01) devices. Added ‘BUTCTL’, ‘CNFREQ’, ‘EPREQ’, ‘LUMCTL’, ‘SPAOPC’, and ‘TESTOP’ opcodes. SFTPT1 - 56:Removed product specific details. These are described per product in separate documents. Added the description of bi-directional reset that was inadvertantly removed in a previous version. RADRDY: Added the requirement to send an additional RADRDY message each time an option is plugged in (after it sends it’s PRUPST). Also, added a Portable/Mobile bit to the RADRDY message. Horn and Lights relay fields of the DISPLY opcode were apparently wrong in this document. The actual fields have been added. Added new option addresses $1A (Control Host) and $1B (Vehicular Adapters) Made it more clear that the following opcodes are obsolete: CHINFO, MEMFRA, and PWRCTL Added virtual address capability to MEMACS. Added a SETBUT message for use with vehicular adapters. This controls speaker routing, fixed (non-varying) volume levels, and fixed RF power levels from the radio. Added a SETBUT message for use in enabling/ disabling field testing of digital radio’s bit error rate and RSSI samples. Updated Appendix H (Data Sub-Opcodes) for changes made in Data on Trunking. Fixed various references to incorrect section numbers, and typographical errors.

Revision #	Date	Author	Description
Version 5.11			The previous version on the standard cover was listed as 5.01 when it really was intended to be 5.1, therefore this version was listed as 5.11. Changed Channel Scan to Dsp in address assignments. Defined bit D3 in MEMACS . It now combining with D2 defines 4 devices. Added more comments as per the usage of MEMACS opcode. Added RLAP_REQ , WRITE_SER_NUM_REQ , RLAP_REPLY and WRITE_SER_NUM_REPLY SBEP opcodes. Modified ERASE_FLASH_REQ opcode. Clarified READ_DATA_REPLY and CHECKSUM_REPLY opcodes.
Version 5.12			Obsoleted ACPRI1, ACPRI2 and ACTNPL opcodes with the introduction of two new opcodes MODUPD and OPTUPD. Added two new opcodes MODUPD (mode update) and OPTUPD (option update) to be used when needing to refer to more than 255 modes in a radio. Updated Control head types for CNFREQ opcode. Added more sub opcodes for WRITE_SERIAL_NUM_REQ SBEP opcode. Updated alert tone types for the test alert message in Appendix I for sbep.
Version 5.13	08/28/95	Wilker Altidor Jerry Sandvos	Converted Version 5.12 from Interleaf to Frame Upissued version number because of page numbering chages. Made minor grammatical corrections

1 INTRODUCTION

The purpose of this document is to describe the operation and architecture of the SB9600 serial bus. The goal of the bus is to provide a general purpose communications protocol which will allow Motorola products to effectively interface with each other. The flexibility thus achieved will provide compatibility and cost effective solutions to system expansion for Motorola's customers.

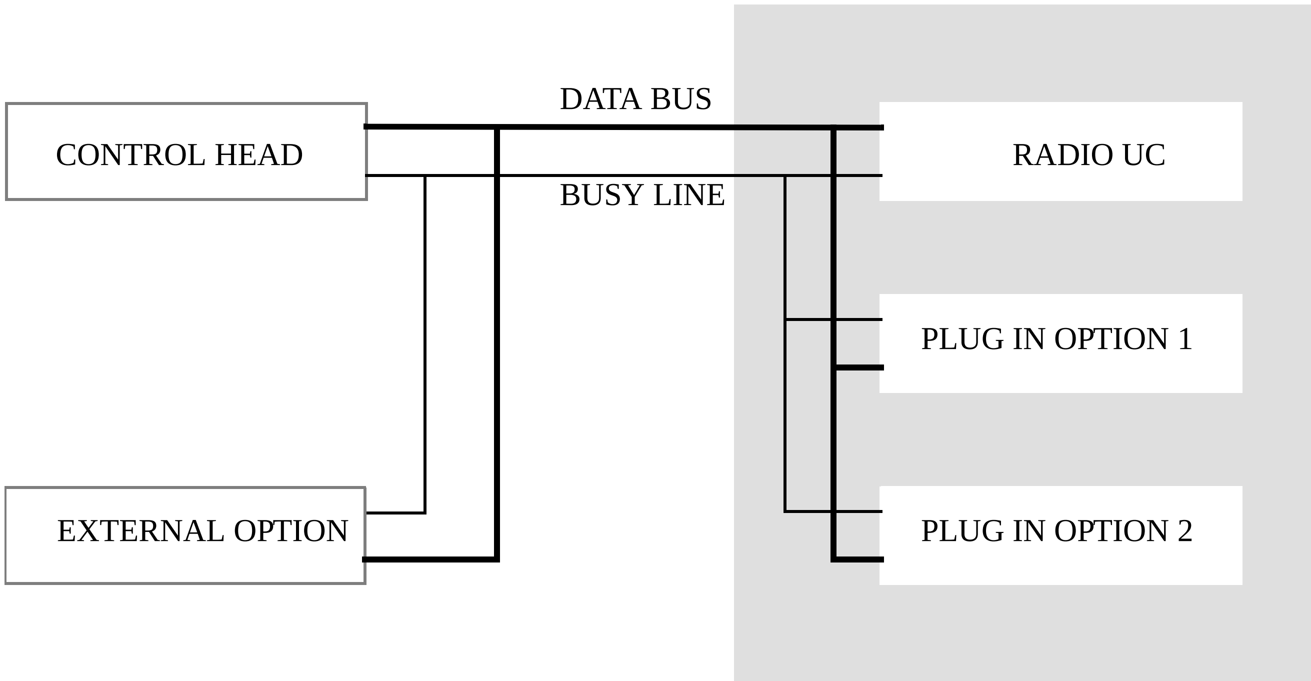
This document describes the full embodiment of SB9600. All applications of the bus should be documented in a supplementary manual to describe particular implementations. Especially important are departures from this document which can take the form of extensions, additions, exceptions, and only partial implementations. Although every effort has been made to foresee future requirements, reality forces allowance of slight departures from the standard described herein. When compatibility and commonality are of concern, the supplements to this document will become just as important as the document itself. Any changes of major importance will be included in updated revisions of this document.

1.1 General Description

The serial bus may be used anytime a communcation need exists amoung processors embedded in a system or between an embedded and external processor(s).

Communications between devices occur using serial data transmitted at 9600 Bits Per Second on a bi-directional data bus. A block diagram depicting one of the original applications of the bus is shown in Figure 1. The architecture is general enough that many different applications can be envisioned.

Access to the various processors in the system is obtained via the bidirectional data line (BUS) and a bidirectional handshake line (BUSY) using techniques similar to CSMA/CD (Carrier Sense, Multiple Access with Collision Detection) which is frequently employed in computer Local Area Networks.



SYSTEM 9000
BLOCK
DIAGRAM

Figure 1-1 - System 9000 Mobile Radio Block Diagram

2 PHYSICAL BUS

The SB9600 serial bus in its original form (the Longhorn bus developed for the SYSTEM 9000 and Spectra mobile radios) described a combination of a serial data bus and a parallel audio, power, and control bus. Because of the greater scope of SB9600 (Consoles, Data Terminals, Portable Subscriber units, etc.) the interface requirements are more diverse and deserving of special attention in a separate document. SB9600 as described here concentrates on the digital serial data interface.

- SIGNAL RETURN (GROUND,)
- BUS
- BUSY

Specific applications may add to these. In a mobile application, for instance, The BUS line may be split into balanced BUS+/BUS- lines to form a complementary signal for noise rejection.

2.1 Audio Routing

The details of the audio interface will be described in a future companion document, however, some definition is appropriate here as the serial bus affects audio paths.

The 5 audio interconnects include:

a microphone (MIC) output,
a discriminator (DISC) output,
a filtered receive audio (FLTAUD) output,
and two general purpose transmit and receive inputs, TXAUD and RXAUD.

These last two lines can be configured as inputs to various points in the system as shown in Figure Figure 2-1. This figure is not intended to be indicative of the method of implementation, but is included merely to illustrate the audio routing and the controls which are available.

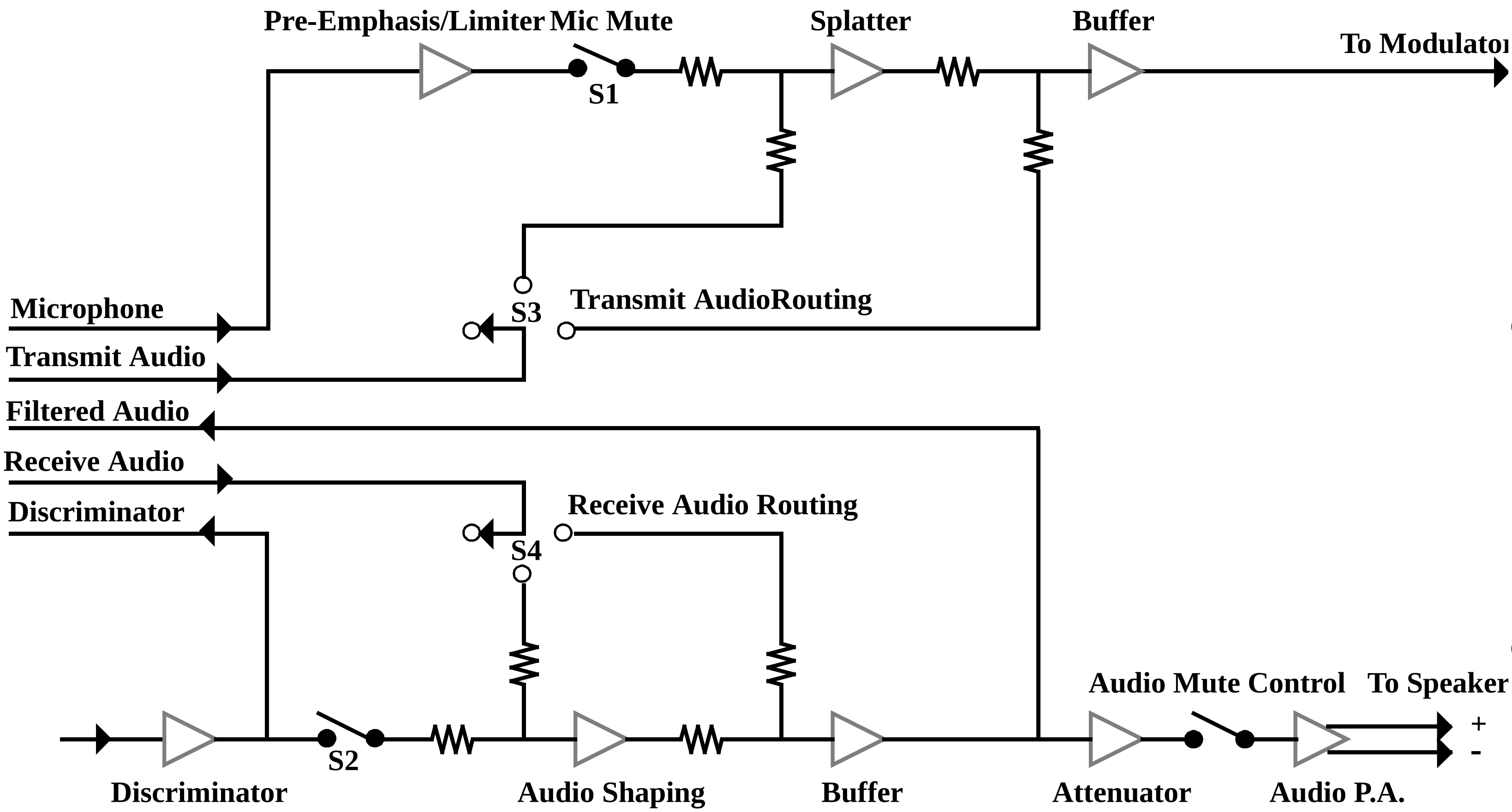


Figure 2-1 - Audio Routing Model

3 BUS PROTOCOL

A device accesses the bus by first checking to see that the BUSY line is inactive (also called ’free’). If it is not free, the device must wait a period of time before trying again. If it is free, it must immediately pull BUSY active, transmit five bytes on the BUS line, and then release BUSY. There are a myriad of special cases, timing constraints, and error conditions that need to be dealt with. These are all discussed in this chapter.

There are two way to communicate information on the bus. The normal method is to use a standard broadcast message or request/broadcast transaction. The second method is used when large amounts of data need be communicated ie. for programming a devices EEPROM. This second method is termed the Serial Bus Expanded Protocol (SBEP).

3.1 Standard Signalling Definition

The basic building block upon which the serial bus is built is the 8 bit data byte. This is sent in the normal fashion with 1 start bit, 8 bits of data (LSB first), and one stop bit. The data byte with the start and stop bits (10 bits total) is referred to as a ‘data packet’. The start bit is a low voltage (’0’) while the stop bit is high (’1’). At 9600 baud, each packet takes 1.042msec to transmit. It is often convenient to equate a ’packet time’ with 1 msec.

Five of these packets are concatenated to form a single ‘message’. Messages typically consist of:

- address packet,
- two data packets,
- an opcode packet,
- and a CRC (Cyclic Redundancy Code) packet.

All standard messages are 5 packets in length. Constant length (five packet) messages are used rather than variable length messages to prevent data corruption which would result if a CRC calculation were performed on the wrong packet due to a loss of message synchronization.

The two types of messages allowed in the system are Requests and Broadcasts. Requests are used to obtain information from other processors in the system. A Broadcast is either a response to a specific Request (solicited Broadcast) or is spontaneously generated and sent to all devices in the system (unsolicited Broadcast), such as when a button is activated.

A ‘transaction’ is defined as a Request followed by a Broadcast response. The ”requestor” maintains control of the bus by keeping the BUSY line active until the first byte of the response is received. A Broadcast does not require a response.

3.1.1 CRC

At the end of all messages is a packet containing an 8 bit Cyclic Redundancy Code for error detection. Typically, after each packet is received, a partial CRC will be calculated. After the last packet, the result should equal 0. This is because the last packet was actually the transmitted CRC. If the locally generated CRC equalled the transmitted CRC, then the

result of performing the exclusive-OR's will be zero. This makes it very convenient for the receivers to determine if the message contained errors or not.

The details of the CRC calculation is not contained in this document. Interested individuals with a "need to know" should contact the World Wide Radio Product Group of the Land Mobile Products Sector for additional information.

3.2 BUSY

The BUSY line is a bi-directional line which mainly serves to indicate when a message exists on the bus. Before a message can be sent on the bus, a device desiring to transmit must first check to see if the BUSY line is active. If it is not active, the device pulls the BUSY line active and sends the message. The time between checking and pulling the line active should be as short as possible (all interrupts disabled between checking and pulling) to minimize collisions. The message must begin within one packet time after the BUSY line has been pulled active. If the BUSY line is found to be active, the device must wait at least 1 packet time and check again until the line becomes inactive. Only in response to a Request should a device begin transmission when another device is pulling the BUSY line is active. Also, it is recommended that devices not sit in a tight software loop waiting for BUSY to be released, as there is a greater probability two such devices will collide. Each device must be 'infinitely patient', meaning that it must wait indefinitely for the busy line to become inactive, and must not timeout and reset the bus, for example.

If a device expects a response to its transmission (i.e. a Request was sent), the BUSY line must be released after receiving the first byte of the responding Broadcast. If a response is not expected (i.e. a Broadcast was sent), the BUSY line must be released between 3 and 4 packet times after sending out the last packet of the message.

The BUSY line is also used to formulate and test for a NAK (see Section 3.6). When a message has a CRC error or if an interpacket delay violation occurs, the receiving device must immediately pull the BUSY line active. After sending the Broadcast, the original transmitter will release and then sample the BUSY line to see if it is still held active, indicating the message was not received correctly. The time between releasing BUSY and then sampling it should be long enough for any cable kit delays to settle (20usec should suffice) but less than 1 packet time. If BUSY is still held active, then the message is being NAK'ed and the original transmitter may send the message again.

It should be noted that during re-transmissions, BUSY never changes state (always active). Once a valid message has been received, all subsequent messages should be ignored until BUSY changes state. This prevents other devices from decoding multiple increment opcodes (eg. TXCNTRL) when a message gets NAK'ed. An exception to this rule occurs during a Request transaction. If a Request has been decoded, the responding Broadcast must still be received, even though BUSY has not changed state. Otherwise, devices may not decode an SBEP Broadcast response and may attempt to execute SBEP data.

BUSY is used to keep devices synchronized. Whenever BUSY goes inactive (or alternatively, whenever it goes active), devices should re-initialize their receive buffers to prepare for a new message. Thus, if a collision takes place (BUSY will go active, a single

byte will be sent, and BUSY released), devices will ignore the single byte and restore their receivers because of the last transition on BUSY (alternatively, the receivers will be restored upon the next BUSY transition to the active state).

BUSY is also used to aid the power up sequence. Upon powering-up, all devices should hold BUSY active until they are ready for normal operation. Even though each device has a different power-up initialization time (due to operating speeds, self-diagnostics, etc.), the system will wait until the slowest device is ready before anything is sent on the BUS line.

During SBEP (see Appendix I), the first message is treated as normal. However, if the SBEP message is meant for another device, then the receiver must be setup to ignore the next message. Since receivers are re-initialized upon BUSY transitions, special action must take place to not allow the device to falsely interpret the following data as a valid opcode.

Timing constraints are such that whenever BUSY goes inactive, it will remain that way for at least 1 packet time. Thus, even though BUSY has edge sensitive properties, it is not necessary to connect it to an edge sensitive input. The system will tolerate BUSY to be polled as long as the polling period is shorter than 1 packet time. A potential danger exists however. It is possible that a collision has occurred between two polling samples, thus loading the receiver with a single byte. This can cause a loss of synchronization unless care is taken to correct this problem. A recommended solution is to also clear the receive buffer. If BUSY is inactive, then the receiver should be re- initialized even if BUSY was previously found to be inactive.

3.3 Timing

The various processors in the system access the bus based on a technique similar to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) employed on various computer Local Area Networks. The timing overhead for this type of bus access is minimal. Typical bus timing is shown in Figure Figure 3-1 and should be referred to when reviewing the following paragraphs.

The data rate on the bus is 9600 bps (+/- 1.875%). It will soon become apparent that a message can appear on the bus no faster than 1 every 9msec. Without SBEP, this corresponds to 1 character every 9msec, or 111 cps. This is only marginally slower than a normal 1200 bps modem (120 cps). With SBEP, the effective rate is increased.

Timing restrictions for the bus will be considered separately for the two types of bus messages: Requests and Broadcasts. Timing is specified in terms of ”packet times” which is the time it takes to send one packet of information (10 bits total) over the bus. This time is 1.042 msec at 9600 bps, although it is more convenient to equate a packet time with 1msec. A packet is defined to be transmitted when it is loaded into the transmit shift register (note that there may be a substantial delay between the time the ’store accumulator’ opcode is executed and the start bit appears on the BUS line). A packet is defined to be received when the receive shift register is full and the byte has been transferred to the receive data register of the microprocessor (again there may be a delay

between the time the stop bit is verified and a ’RX buffer full’ flag is set). All delays inherent in a device (eg. portions of a program where an interrupt is masked, hardware delays described above) must be taken into account when trying to meet the timing restrictions described here. Generally, a device can meet the timing requirements if the delays are less than 800usec per packet time (however, if the device wants to do anything in addition to talking and listening on the bus, the delays must be even less).

A Request is transmitted to initiate a transaction whenever a device desires information. The response is always a Broadcast. To start a Request, BUSY is checked, and if found inactive, it is immediately pulled active (in order to minimize collisions, the time between checking BUSY and pulling it active should be as small as possible - ie. interrupts should be disabled). The first packet must then be transmitted within one packet time of BUSY going active (A). Each of the four subsequent packets must then be transmitted within one packet time after the previous packet was received (interpacket delay time (B)). Note that since the BUS line is bi-directional, a transmitting device will receive its own transmission. Once the Request has been completely transmitted the BUSY line must be held active until the first byte of the response (a Broadcast) has been received (D). If the first byte of the response is not received within 5 packet times (C), BUSY is released and the Request may be attempted again later. It should be noted that the length of a message can vary anywhere from 5 packet times to 9 packet times.

A device responds to the Request by sending the appropriate Broadcast and must return the first byte of the response within 4 packet times of receiving the last byte of the Request (C). The response, like the Request, has a maximum interpacket delay time of one packet. The responding device should release BUSY and check for a NAK between 3 and 4 packet times after receiving its own Broadcast (E). Since the delay between the Request and Broadcast can vary from 0 to 4 packet times, the length of the entire transaction can vary from 5+0+5+3+1=14 packet times to 9+4+9+4+1=27 packet times.

A Broadcast is transmitted as a response to a Request, as a command to another device (eg. RXAUD), or to announce a change of state of an internal parameter (eg. RADKEY). The Broadcast is started just like a Request and has the same interpacket delay time (see Figure Figure 3-1). The difference is that Broadcasts are not responded to. The device sending the Broadcast releases BUSY and checks for a NAK 3 to 4 packet times after receiving his own Broadcast. A NAK occurs when a device on the bus pulls BUSY active after sensing a CRC error (F) or an interpacket delay time violation (an interpacket delay timing violation is considered to have occurred when more than 2 packet times pass between packet receptions (B)). Since the NAKing device pulls BUSY active, the responding device senses this and sets up to retransmit the Broadcast (G). The NAKing device should release BUSY when the first packet of the re-Broadcast is received (H), or after 5 packet times after initiating the NAK (I).

\When waiting to use the bus, devices should monitor BUSY. If BUSY has remained inactive for 1 packet time or more, then the bus is considered available. The total length of a Broadcast includes the actual message length, NAK time, and the packet time of inactive

BUSY. This varies from 5+3+1=9 packet times to 9+4+1=14 packet times. The bus is therefore 5/9=56% efficient when fully loaded.

If a bus collision is detected (see Section 3.6) the detecting device(s) must immediately release BUSY and wait at least 1 packet time before starting the message over again. In multiple radio systems, it is possible to have the same type of option on the bus more than once (each option with its own GROUP ADDRESS, see Section 3.7). In order to keep these from continually colliding with each other, it is recommended that they increase their wait by approximately (GROUP/DEVICE ADDRESS)/data rate.

Timing during SBEP is described in Appendix I.

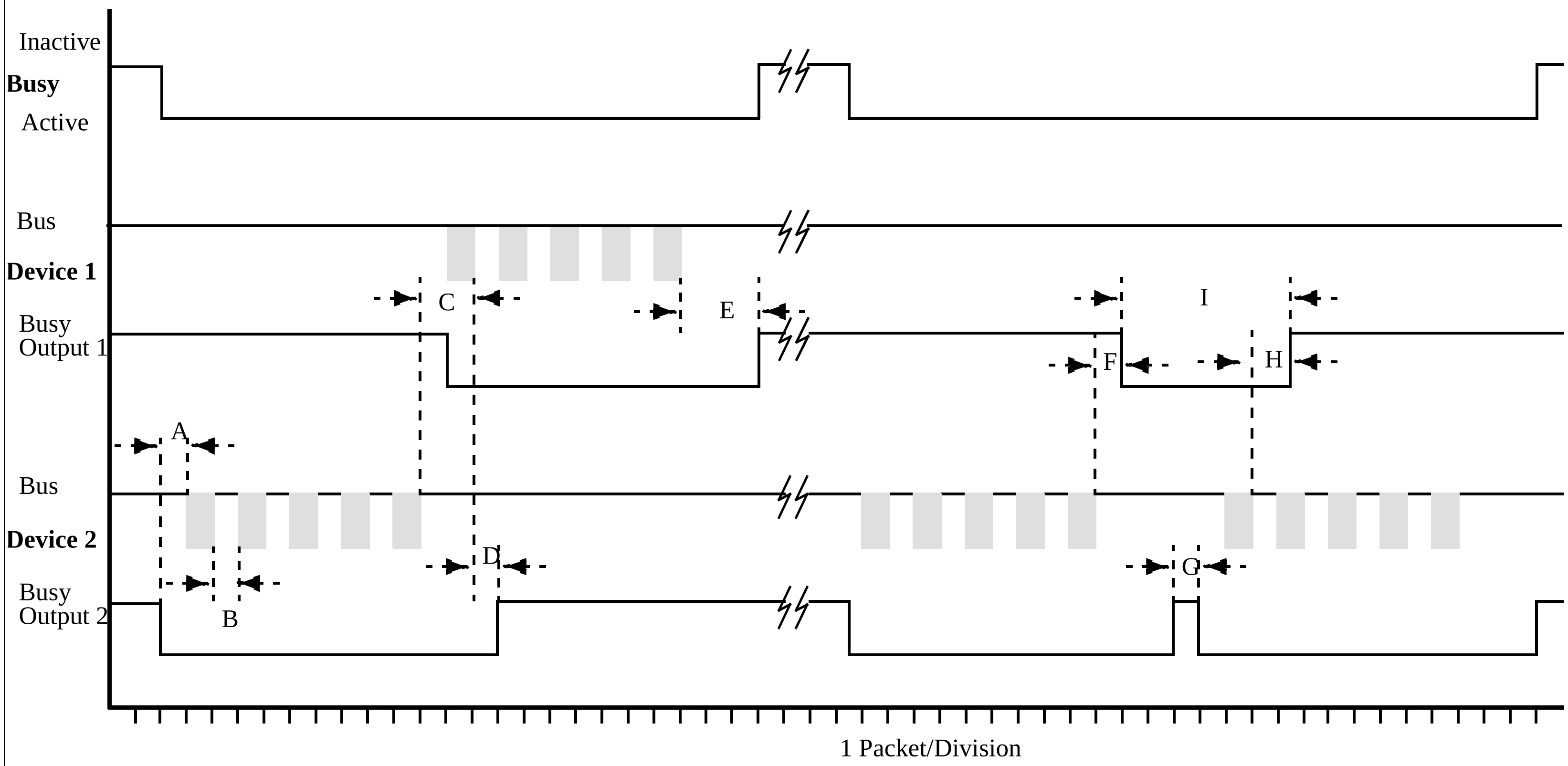


Figure 3-1 - Bus Timing Diagram for Standard Messages

A. After Busy goes active, the first packet is transmitted within 1 paket time

B. No more than 1 packet time between packets (bytes)

C. Broadcasts follow requests within 5 packet times

D. Busy is released within 1 packet time after first byte of response

E. Busy is released and checked for a ”NAK” between 3 and 4 packet times after the broadcast is received

F. Device 1 ”NAKs” Device 2 by pulling busy active within 1 packet time of received broadcast

G Busy is pulled active (low) within 1 packet after Device 2 recognizes that it was ”Naked”

H. Busy is released within 1 packet time after the first packet (byte) of rebroadcast

I. ”NAKing” Device should hold busy active at least 5 packet times if no re-broadcast is received

3.4 Power Cycling

3.4.1 Power Up

During power up, certain special actions take place. Each device holds BUSY active until they are ready to receive data on the BUS line. This time interval allows devices to perform their power up routines and guarantees that no data will be sent until the slowest device is ready. Once BUSY becomes available, each device must send their PRUPSTS. Before sending its PRUPSTS, the control head also sends SETBUT (older devices) or BUTCTL (newer devices) to indicate the state of any VIP and switch inputs and OPTSTS Requests to get any display or button mapping from the radio. The radio ends the power-up interval by sending RADRDY. Up until RADRDY is sent, the radio will mute the audio and will not key-up. Also, before RADRDY is sent, the bus can only be used:

1) for sending PRUPSTS,
2) for responding to any Requests,
3) by the control head when sending initial VIP/switch states and OPTSTS Requests (see note below).

Note: It is recommended that all OPTSTS requests be sent ***after*** RADRDY is sent. The only exception is when very time sensitive information is required, and even then, only a single OPTSTS request should be sent (postpone remaining ones, if any, until after RADRDY is received). The reason for this is that if a lot of these messages are generated, RADRDY will be delayed, and the control head(s) can time out while waiting for RADRDY, and reset the system.

After RADRDY, the bus is available to all devices in the normal fashion.

The radio doesn’t send RADRDY until:

1) each device has sent its PRUPSTS or,
2) until a system power-up timer has expired.

If 1) is satisfied, then RADRDY is sent as described above. If 2) is satisfied, then the radio must send the PRUPSTS for each missing option indicating that there has been a hardware failure (bit 4 of PRUPSTS) and whether or not it is considered a fatal error (bit 7 of PRUPSTS). After, these are sent for each missing option, RADRDY is sent (assuming none of the PRUPSTS had a fatal error). Thus, the radio must support some additional EEPROM. The timer value should be flexible enough to allow for the increased power-up delays incurred by larger, multi-radio systems. Also, the radio needs to know what addresses are supposed to be occupied on the bus and whether or not a missing option is considered a fatal system error.

Typically, devices will go thru a power-up diagnostic mode where RAM, ROM, and EEPROM are verified. They will then Broadcast their Power Up Status. If at any time, a PRUPST is Broadcast indicating there is a fatal error, the control head will indicate this on the display and lock out the keypad. Also, the radio will mute audio and de-key the RF transmitter.

If a device finds its EEPROM is blank, it should only send a PRUPST (a blank EEPROM should be considered a fatal error). It shouldn't transmit anything else on the bus until a MEMACS has been received with its address.

3.4.2 Power Down

There is no way to predict when power will be removed from the system. If options need to remember status during a power cycle, they should use an unswitched 5v line to keep their RAM active when the radio is turned off. They have the burden to ensure that they properly power themselves down with RAM intact. In order to reduce power-up delay time, it is recommended that devices remember Option Status in RAM. Thus, it will be requested only when power is removed from the system (eg. system installation).

3.5 The Human Interface

3.5.1 Buttons

In a mobile subscriber environment, control head buttons are the primary means the operator has to manipulate and use his radio system. Since various control heads and keypad entry devices will be utilizing the SB9600 serial bus, every effort has been made to make the interface with the various devices on the bus independent of the peculiarities of keypad hardware.

Each device will not support its own hardware switches, but will remotely share a common keypad with other devices. Thus, when the operator presses a button, the control head must so inform the particular option. There is no need for options to poll the state of a switch since they will be instructed when a button for them has been pressed. In fact they are discouraged from doing so since it would needlessly tie-up the bus. The control head uses a series of BUTTON opcodes to instruct the options when a button has been pressed. These include:

BUTPRS, DECBUT, DELBUT, ENTBUT, EXTBUT,
INCBUT, RCLBUT REQBUT, SETBUT, SHOBUT.

Each of these opcodes refer to a specific type of button) and can be applied to various aspects of an option's operation (eg. the PA volume level of a SIREN option can be decremented as can its mode of operation). The particular aspect of an option's operation is specified in the BUTTON REGISTER and may also include some DATA. This is sent by the control head and is part of the BUTTON opcode. Some BUTTON REGISTERS are actually switch closures or ports for obtaining vehicle status and exist in all radio systems (eg. the PTT switch) while others are option specific (eg. DVP on/off). These general system BUTTON REGISTERS and their associated DATA are shown below and will be sent to GROUP/ADDRESS $00, usually with SETBUT.

BUTTON REGISTER	OPERATION	DATA
$00	POWER SWITCH	0/1 1=power ON
$01	HUB/ERGO SWITCH	0/1 1=ON hook (muting enabled)
$02	IGNITION SWITCH	0/1 1=ignition ON
$03	PTT	0/1 1=PTT pressed
$04	EMERGENCY	0/1 1=button pressed
$05	HORN RING	0/1 1=horn ring pressed
$06	for future use	-
$07	for future use	-
$08	for future use	-
$09	for future use	-
$0A	for future use	-
$0B	for future use	-
$0C	for future use	-
$0D	for future use	-
$0E	for future use	-
$0F	for future use	-

As options are developed, new BUTTON REGISTERS will be defined to accommodate that option. Thus, expandable control heads must have programmable, non-volatile storage to map these new BUTTON REGISTERS onto a particular button. Each option must list their BUTTON REGISTERS in an accompanying supplement. A listing of available BUTTON OPCODES will be contained in supplements describing the various control heads (some may implement the entire set while others may only transmit SETBUTs). APPENDIX E contains a master listing of button registers as of the date of the Appendix.

The ENTBUT and EXTBUT do not necessarily represent physical buttons, but may be sent as a result of a sequence of button presses. These are sent to cause a device to halt its normal activity and to enter a configuration mode, allowing the user to modify certain operating parameters (eg. setting up a scan list, choosing an MPL code, dialing a phone number, etc). When the operator has completed his manipulation, the EXTBUT will be sent with DATA indicating the last selected state of the parameter. Some devices may not need this information.

EXTBUT can also be sent by options to force the control head to exit configuration based on some non-user initiated event. It should be noted that the configuration mode is not exited until the control head itself sends EXTBUT.

If the user attempts to exit the configuration mode of an option which needs to validate the modified parameters, the control head sends REQBUT, requesting that the option exit

configuration. Upon validation, the option sends an EXTBUT to the control head, instructing it to exit. The control then completes the sequence by sending its own EXTBUT.

The SETBUT is sent to a device with DATA indicating that a new value has been selected for that BUTTON REGISTER. An example is when a user selects a new mode. If the control head has mode push buttons, it will send SETBUT with DATA indicating which mode was selected. Alternatively, if the control head has a scrolling up/down mode control button, it will send INCBUT and DECBUT without DATA.

Since some (most) control heads cannot show all available information simultaneously, a SHOBUT is available to allow the operator to see a selected state. The response to a SHOBUT directed to a particular GROUP/ADDRESS and BUTTON REGISTER will be a DISPLY indicating the current value of the BUTTON REGISTER for that device.

A BUTPRS is sent when ASCII type data appears on the bus. This occurs when text or a variable length number is to be input. An example is a telephone number. As each digit is pressed, the number (in ASCII) is put on the bus with BUTPRS. This allows the option to do its own editing. The alternative is to let the control head do its own local editing and send the result down as a SETBUT. However, the number thus obtained must be less than 256. Either option is available and must be specified in the option and control head supplementals.

The RCLBUT and DELBUT allow the operator to edit lists. The primary example is channel scan. Usually these will only be sent while the device is in its configuration mode.

3.5.2 Displays

Displays are the primary means a radio system has to communicate status and information about the system. Since various control heads and display devices will be utilizing the SB9600 serial bus, every effort has been made to make the interface with the various devices on the bus independent of the peculiarities of display hardware.

Each device does not support its own hardware display, but will remotely share a common display with other devices. Thus, when a device wishes to have something shown on the display, it must so instruct the control head. This is done with the DISPLY opcode. Included with this opcode are FIELD and DATA bytes. The FIELD byte can be thought of as defining a location or function on the display while the DATA byte describes what is written there. Some FIELDS exist which control ports to the vehicle and are available to any option for manipulation. These general system FIELDS and their associated DATA are shown below. Devices write to these using GROUP/ADDRESS $00 allowing a device to monitor how other devices are manipulating the port). All other fields ($10 to $FF) use the normal source GROUP/ADDRESS and are used by particular devices.

FIELD	LOCATION	DATA
$00	HORN RELAY ?	0/1 1 = energize
$01	LIGHTS RELAY ?	0/1 1 = energize
$02	HORN RING RELAY	0/1 1 = energize
$03	for future use	-
$04	for future use	-
$05	for future use	-
$06	HORN RELAY	0/1 1 = energize
$07	LIGHTS RELAY	0/1 1 = energize
$08	for future use	-
$09	for future use	-
$0A	for future use	-
$0B	for future use	-
$0C	for future use	-
$0D	for future use	-
$0E	for future use	-
$0F	for future use	-
$10 - $EF	for future use	-
$F0 - $FF	for future use	ASCII Display Fields

In addition to these, FIELDS $F0 thru $FF are reserved to represent various ASCII display fields. In this case, DATA represents ASCII data and can range from $00 thru $7F. It is entirely possible for every device to have its own block of 16 ASCII display areas (assuming the control head supports it).

As new devices are developed, FIELDS will be defined to accommodate them. Thus, expandable control heads must have programmable, non-volatile storage to map these new FIELDS onto their particular display. Each option must list their FIELDS in an accompanying supplement. A listing of supported FIELDS and how these FIELDS appear physically on the display will be contained in supplements describing the various control heads. APPENDIX F contains a master listing of display FIELDS as of the date of the Appendix.

3.5.3 Messages

Messages are similar to displays except that they are 'latched' at the control head. An example is the 'CALL' light of a data signalling option. The CALL should be always showing until the operator presses some button, or takes some other action, indicating to the option that the CALL has been properly dealt with. Since the display is remote from

the option, the control head must be informed when a message is to appear and when it is to be released. This is accomplished by the DSPMSG opcode and uses the same form as DISPLY. This informs the control head to 'latch' the specified FIELD/DATA onto the display. Once the option has decided that proper action has taken place, it may remove the message from the FIELD by writing DSPMSG with the same FIELD and DATA = 0. Different control heads will have different message capabilities (multiple messages, single message w/queueing, no queueing, etc) and will be specified in the control head supplement.

3.6 Error Handling

Errors occur due to outside noise sources or improper system utilization. Errors in data transmission are detected by the CRC (see Section 3.1.1) or by timeout (see Section 3.3). When such an error occurs, a NAK is sent. Note that errors reported by UARTs (eg. overrun and framing errors) can usually be ignored.

A NAK consists of pulling the BUSY line active during the "NAK period" following a Broadcast. The mere presence of an active BUSY in this time interval indicates that some listener did not receive a valid CRC. Even if several devices simultaneously pull BUSY active (to NAK), the event will still be understood since they are all "wired OR'ed" together. A NAK will cause any Broadcast to be retransmitted. A NAK should be aborted (BUSY released) if data appears on the BUS line or after 5 packet times.

Errors in execution of control messages are detected only upon attempting the execution. It is the responsibility of the devices to guarantee that they follow the recommended procedures outlined in this document. If this type of error does occur (i.e. a Request TX Mode while the radio is programming the EEPROM) one of the following three things will happen:

1) nothing,
2) response may be meaningless,
3) the system may be reset.

3.6.1 CRC Errors

Noise on the bus may cause bit errors in the data stream. These are detected by the CRC (see Section 3.1.1). If a receiver detects a CRC error, it immediately NAKs by pulling BUSY active. Since the receiver doesn't know when the transmitter will check for the NAK, it must hold it active until 1) another data packet has been received, or 2) 5 packet times have elapsed, whichever occurs first. Once one of these requirements is met, BUSY must be released. The time when BUSY can be held active is called the 'NAK period'.

After a transmitter has finished sending its message, it must continue to hold BUSY active for at least 3 more packet times to give receivers time to process the CRC. It then releases BUSY, waits an appropriate delay (20usec is sufficient for a typical cable kit delay) and reads the BUSY line to see if it is still being held active. If so, a NAK is registered and the device immediately retransmits the entire message. Under no circumstance must it hold BUSY for more than 4 packet times or receiving devices may release BUSY before the transmitter has checked it. Also, the time between releasing BUSY and checking for NAKs

must be less than 1 packet time. Otherwise, a third party may find BUSY available and pull it active causing the original transmitter to falsely read a NAK.

Upon detecting a NAK, the transmitting device may retransmit the message. Upon detecting further NAKs, the device should do one of the following:

1) try again up to 8 times total,
2) release BUSY, wait at least 1 packet time and try again,
3) release BUSY, wait at least 1 packet time, and send PRUPST with bit5 set,
4) RESET the system.

3.6.2 Synchronization Errors

If devices lose synchronization with each other, then a CRC calculation will be made and compared with random data, not the received CRC. The probability of error obtained (.4%) is too great to allow this condition to exist for an extended length of time. The worst case condition, the radio inadvertently keying up, is sufficiently undesirable that special precautions need to take place to keep this from happening.

While transmitting, each packet must follow the previous packet within 1 packet time. While receiving, each device must set up a 2 packet time timer. Each time a packet is received, the timer is started over. Upon receipt of the last (5th) packet, the timer is disabled. If the timer ever expires, then an ’interpacket delay violation’ has occurred and the receiver must immediately NAK (the transmitter is probably in its own ’NAK period’). This NAK is released when:

1) another data packet has appeared on the BUS line,
2) 5 packet times have elapsed, in the normal manner described above.

Any time an interpacket delay violation occurs, the partially received data must be disregarded and the receiver re-initialized to receive a new message.

3.6.3 Collisions

It is possible that two (or more) devices may want to use the bus at the same time. This is detectable by the transmitting devices since the bi-directional BUS line forces the receive and transmit lines to be tied together. Thus a device can read back what it sent out. Each time a packet is to be sent out, the receive serial buffer is checked to see if it matches the previously sent data. If they do match, then no collision has occurred and transmission can proceed as normal. If it does not match, then a collision has occurred. It should be noted that a collision is detected only after a whole packet is sent and thus third party receivers (who don’t know a collision has occurred) will be waiting for more data.

When transmitting device has detected a collision, it immediately stops transmitting, releases BUSY, goes to receive, and must wait at least 1 packet time before trying again. Other than the 1 packet minimum wait time, there are no restrictions imposed on the delay time before retry can occur. However, since each device has a natural loop time associated with it, it is recommended that devices use this time as a natural retry delay. Since these delays are variable and unsynchronized, it is unlikely that another collision will occur. It is recommended that devices do not wait in a ‘tight loop’ for BUSY to become inactive, since the two devices would then have a higher probability of colliding

again. In certain applications several devices may be somewhat synchronized. In such cases, devices should wait a random time before retry.

Receivers on the bus cannot tell if a collision has occurred. They are kept synchronized by the state of the BUSY line. Each time BUSY goes active (alternatively they can look for the occurrence of BUSY going inactive), they should clear their receive buffers and re-initialize to receive a new message. Thus, during a collision, BUSY would be pulled active, a single packet sent, and BUSY released. This causes all receivers to clear out the single packet they received.

3.6.4 Errors During a Request

Since NAKs are not checked during a Request-Broadcast transaction (BUSY is never released, see Section 3.3), a Request will never get NAK'ed. If the first packet of the response is not received within 5 packet times, the device may release BUSY, wait at least 1 packet time, and try the Request again later. Upon a further lack of response, the device should do one of the following: 1) release BUSY, wait at least 1 packet time, and try again up to 8 times total, 2) release BUSY, wait at least 1 packet time, and send PRUPST with bit5 set, or 3) RESET the system.

Collisions are handled normally. If a collision occurs during the Request, the transmitting device should stop transmitting, release BUSY, go to receive, and wait at least 1 packet time before trying again. If the collision occurs during the Broadcast, it should do the same (note that this should never happen since BUSY was held active long before the Broadcast was even started). However, if the original Requesting device does not get its expected opcode as its next response, or if BUSY is released before the complete message is received, then it should send its Request again. It is not necessary for the responding device to retransmit due to a collision (however, a responding device should still respond to NAKs just as if it were sending a normal broadcast).

3.6.5 Being Blind to the Bus

If a device has been blind to the bus for a while and returns finding the bus is BUSY, it must ignores all data transmissions until BUSY goes high. This is because it will not be synchronized with the rest of the system (e.g. is this packet a CRC, opcode, or data?). However, it is recommended that devices never be blind to the bus for extended periods of time. Not only will they be operating with potentially old data (e.g. a Broadcast was sent indicating a Mode change but the device did not receive it), but they may miss an SBEP message and still think the bus contains valid opcode messages (potentially treating raw data as a valid opcode). Generally, all bus timing constraints can be met if the device is blind to the bus (i.e. SCI interrupts disabled) for approximately 800usec every 1 msec (although if the device wishes to do something else besides listening and talking on the bus, the allowable blind time gets smaller). Also, even though much of the error recovery is based on BUSY transitions, the system does allow for a device to poll the BUSY line (as

opposed to connecting it to an edge sensitive input) as long as it is polled more than once every packet time.

An innovative device might resort to holding BUSY active while it is blind to the bus, prohibiting further activity on the bus. Although this is not expressly prohibited, it can tremendously slow down the response of the bus. In using BUSY in this manner, it is also possible that it pulled BUSY simultaneously with another option. The blind device would effectively generate continual NAKs, potentially causing the other device to send PRUPSTS with bit 5 set and resetting the system.

3.6.6 Self-Diagnostics

During power up, each address will perform certain checks to see if they are operating correctly. If an error occurs, the Power Up Status Opcode is sent indicating the location and type of error. Also, if no error occurs, the opcode is sent indicating everything is OK. If any fatal error is Broadcast, the radio will sound an alert tone and the control head will lock out the buttons and write an error message to the display. Also, both the control head and radio will hold BUSY active, thus locking up the bus.

3.7 Addresses

Each option, control head and radio in the system has assigned to it a group and device address. No two options can have the same device address and group. However, one option can occupy several device addresses. SB9600 will support up to 31 device addresses and 7 groups.

Three bits of the first byte of the message are used for group address. Group zero (000) is defined as all groups (i.e. if group 000 is received then the device decodes it as his own group, and if a device is assigned group 000 it decodes all messages sent regardless of the ”sending” group). Group addresses are sent out on most messages (exception: MEMADD, and MEMFRA). On each transmission, devices send their own group, and on each reception, the devices decode the group, and if it does not match their own and is not zero, they must ignore the message (exception: MEMADD, MEMFRA, and all SBEP opcodes). Groups are used to allow separate radio/option systems to operate on the same bus. This is especially useful in multiple radio applications (see Section 4). For single radio systems all devices are typically placed in group 000.

Device addresses are sent out with a message when it is necessary to specify a particular device, such as when the control head sends a SETBUT opcode. Device addresses are also used to read Option Status in the radio EEPROM. Five bits of the first byte of a message are allowed for device address definition. Device address zero (00000) is similar to group zero (000) in that all devices should read it as their own address. For example, a SETBUT with device Group/Address $00 (eg. PTT) is processed by all options, radios and control heads. CHLNUM, MEMADD, MEMFRA, TIMEUP, DATCHN, DATOPC, and SYSTAT do not have Device addresses associated with them.

These addresses, as well as Group assignments, should be read in on power up from an on board source (i.e. jumpers or local EEPROM) that can be easily changed. Certain system

configurations may require these assignments to vary. If this is not possible, then care must be taken in choosing the fixed address in order to minimize option incompatibilities.

For additional information on RX/TX audio characteristics of some of the above devices, refer to Appendix C.

Address assignments have been reserved as follows:

ADDRESS	DEVICE
$00	ENTIRE SYSTEM
$01	RADIO
$02	DSP
$03	MPL
$04	INTERNAL RADIO OPTIONS
$05	FRONT CONTROL HEAD
$06	REAR CONTROL HEAD
$07	CONTROL HEAD EXTENSIONS
$08	Siren/PA
$09	Securenet
$0A	Emergency
$0A	Status/ID
$0B	Message
$0B	MDC1200 SelCall
$0C	MDC600 SelCall
$0D	MVS Emergency
$0D	MVS ID
$0D	MVS Audio
$0E	Phone
$0F	DTMF
$10	Trunking System
$11	Trunking Options
$12	Vehicular Repeater
$12, $13	SP Repeater
$14, $15	Single Tone
$16	Vehicle Location
$17	KDT Terminal
$18	Trunked deskset
$19	Metrocom
$1A	Control Host (see note)
$1B	Vehicular Adapters
$1C	available
$1D	available
$1E	available
$1F	available

Note: Examples of ‘Control Hosts’ ($1A) are Radio Service Software (RSS), Smart Radio Interface Box (SRIB), and automatic test systems.

3.8 Opcodes

Opcodes are used to instruct options, the control head, and the radio to perform certain actions or how to interpret subsequent data. They usually affect corresponding RAM locations in the device processor and many of them control the same functions the operator has access to via the Control Head. The first byte of a message is the GROUP and ADDRESS of the source device (exceptions are MEMACS, MEMADD, MEMFRA, OPTSTS, PRUPST, DEVJSR, PRTVAL, and all BUTTON opcodes) as described in Section 3.7 The next two bytes are either data or address information. The fourth byte is the opcode and consists of 7 bits of opcode type (allowing up to 128 opcodes) preceded by a bit which indicates whether it is a Broadcast opcode ('0') or a Request opcode ('1'). These opcodes are defined in Section 3.8.5 The fifth byte is the CRC and is described in Section 3.1.1

3.8.1 Opcode Extensions

Opcodes not yet defined can be reserved for use by multi- processor options to allow them to talk to each other. An example might be a LORAN C vehicular location device made up of two processors. The two processors can communicate with each other using newly defined opcodes which only they understand. This allows them to use the existing serial bus to communicate instead of supporting an additional dedicated link. SGINFO is an example of an opcode dedicated to an option. An alternative to defining new opcodes is to use MEMACS and write the information directly into the option's memory with MEMADD or MEMFRA. A third method to pass information is to use BUTTON opcodes and write data into a button register.

A natural extension of the existing opcode set is to expand the Request opcodes. The only consequence of this is backward compatibility since initial products will not support these. Thus, as an example, an option that requires a Request SQLDET opcode will not be useable with earlier SB9600 products.

It becomes obvious that any modifications or extensions to the existing opcode set must be fully documented and included as an appendix or revision of this document.

3.8.2 Alphabetical Opcode Table

MNEMONIC	OPCODE Name	Request Code (Hex)	Broadcast Code (Hex)
ACDADJ	ACTIVE DEVIATION ADJUST	--	41
ACNPLB	ACTIVE NP LIST BLANKING	--	38
ACPRI1	ACTIVE PRIORITY 1 MODE	--	39

MNEMONIC	OPCODE Name	Request Code (Hex)	Broadcast Code (Hex)
ACPRI2	ACTIVE PRIORITY 2 MODE	--	3A
ACPRVL	ACTIVE TX PWR VALUE	--	33
ACRXPL	ACTIVE RECEIVE PL CODE	--	29
ACTMDU	ACTIVE MODE UPDATE	--	1F
ACTMDW	ACTIVE MODE WRITE	--	2F
ACTNPL	ACTIVE NP LIST	--	37
ACTSTU	ACTIVE STATE UPDATE	--	49
ACTXPL	ACTIVE TRANSMIT PL CODE	--	2A
ALARMS	ALARMS	--	47
ALRTTN	ALERT TONE	--	1C
AUDMUT	AUDIO MUTE	--	1D
BATTST	BATTERY STATUS	--	3E
BUTCTL	BUTTON CONTROL	--	57
BUTPRS	BUTTON PRESS	--	14
CHINFO	CHANNEL INFORMATION **(OBSOLETE)**	81	01
CHLNUM *	CHANNEL NUMBER	--	3F
CNFREQ	CONFIGURATION REQUEST	--	59
DATCHN *	DATA CHANNEL	--	4C
DATOPC *	DATA OPCODE	--	4B
DECBUT	DECREMENT BUTTON	--	0C
DEFBUT	DEFINE BUTTON	--	55
DELBUT	DELETE BUTTON	--	0F
DEVJSR	DEVICE SUBROUTINE JUMP	--	17
DEVVAL	DEVIATION VALUE	--	32

MNEMONIC	OPCODE Name	Request Code (Hex)	Broadcast Code (Hex)
DISMUT	DISCRIMINATOR MUTE	--	22
DISPLY	DISPLAY	--	3C
DSPMSG	DISPLAY MESSAGE	--	3D
ENTBUT	ENTER CONFIGURATION	--	0D
EPREQ	EXPANDED PROTOCOL REQUEST	--	06
ETONOF	EXTENDER ON/OFF	--	35
EXTBUT	EXIT CONFIGURATION	--	0E
INCBUT	INCREMENT BUTTON	--	0B
LUMCTL	ILLUMINATION CONTROL	--	58
MEMACS	MEMORY ACCESS	--	08
MEMADD **	MEMORY ADDR./DATA	87	07
MEMFRA	MEMORY ADDR./DATA FRAMED **(OBSOLETE)**	83	03
METINF	METROCOM INFORMATION	--	4F
OPTSTS	OPTION STATUS VALUE	96	16
OSCVAL	OSCILLATOR VALUE	--	43
PALIM	PA LIMIT VALUE	--	42
PLDECT	PL DETECT	--	23
PRTVAL	PORT VALUE	--	44
PRUPST	POWER UP STATUS	--	3B
PTTINH	PTT INHIBIT	--	18
PWRCTL	POWER CONTROL **(OBSOLETE)**	--	48
RADKEY	RADIO KEYED	--	19
RADRDY	RADIO READY	--	15

MNEMONIC	OPCODE Name	Request Code (Hex)	Broadcast Code (Hex)
RCLBUT	RECALL BUTTON	--	10
REQBUT	REQUEST BUTTON	--	11
REVINH	REVERSE BURST INHIBIT	--	2C
RPTDIR	REPEAT/DIRECT	--	24
RXAUD	RECEIVE AUDIO ROUTING	--	1A
RXMODE	RECEIVE MODE UPDATE	A1	21
RXPLIN	RECEIVE PL MUTE CONTROL INHIBIT	--	2B
SCONOF	SCAN ON/OFF (ALL SCANS)	--	2D
SETBUT	SET BUTTON	--	0A
SFTPT1 *	SOFT POT REGISTER #1	D6	56
SGINFO	SIGNALLING INFORMATION	A5	25
SHOBUT	SHOW BUTTON	--	09
SPAOPC	SPECIAL APPLICATIONS OPCODE	--	5A
SQLDET	SQUELCH DETECT	--	1E
SQLVAL	SQUELCH VALUE	--	28
TBONOF	TALKBACK ON/OFF	--	31
TESTOP	TEST OPCODE	--	5B
TIMEUP *	TIME UPDATE	--	4E
TOTVAL	TIME-OUT-TIMER VALUE	--	36
TRKMDU	TRUNKED MODE UPDATE	--	46
TRKOPC	TRUNKING OPCODE	--	45
TSTMOD *	TEST MODE	--	40
TXAUD	TRANSMIT AUDIO ROUTING	--	1B
TXCTRL	TRANSMIT CONTROL	--	2E

MNEMONIC	OPCODE Name	Request Code (Hex)	Broadcast Code (Hex)
TXLTIN	TRANSMIT LIGHT INHIBIT	--	34
TXMODE	TRANSMIT MODE UPDATE	A0	20
UNQSCN	UNQUALIFY SCAN	--	30
VOLMIN	VOLUME MINIMUM	--	26
VOLVAL	VOLUME VALUE	A7	27
VRSSTU	VEHICULAR REPEATER STATUS UPDATE	--	4A
XFRBDY	TRANSFER DATA BODY OF SEQUENCE	--	51
XFREND	TRANSFER DATA END OF SEQUENCE	--	52
XFRERR	TRANSFER DATA SEQUENCE ERROR	--	53
XFRSTR	TRANSFER DATA START OF SEQUENCE	--	50
XTLPUL	PROCESSOR CRYSTAL PULL	--	54

* Messages containing these opcodes do not have a device address field, and are therefore very specific and limited. They cannot be used in systems that would result in ambiguity due to the lack of the device address information.

** Group and Address are set up by MEMACS.

3.8.3 Functional Opcode Table

FUNCTION	MNEMONC	RQST/BCST CODE (Hex)	
TRANSMITTER			
	ACDADJ	--	41
	ACTMDU	--	1F
	ACTMDW	--	2F
	ACTSTU	--	49
	ACPRVL	--	33
	ACTXPL	--	2A
	DEVVAL	--	32
	OSCVAL	--	43
	PALIM	--	42
	PTTINH	--	18
	PWRCTL	--	48 **(OBSOLETE)**
	RADKEY	--	19
	REVINH	--	2C
	RPTDIR	--	24
	TBONOF	--	31
	TOTVAL	--	36
	TRKMDU	--	46
	TXAUD	--	1B
	TXCTRL	--	2E
	TXLTIN	--	34
	TXMODE	A0	20
	XFRSTR	--	50
	XTLPUL	--	54

FUNCTION	MNEMONC	RQST/BCST CODE (Hex)	
RECEIVER			
	ACRXPL	--	29
	ACTMDU	--	1F
	ACTMDW	--	2F
	ACTSTU	--	49
	ALRTTN	--	1C
	AUDMUT	--	1D
	DISMUT	--	22
	ETONOF	--	35
	OSCVAL	--	43
	PLDECT	--	23
	PWRCTL	--	48
	RXMODE	A1	21
	RXPLIN	--	2B
	RXAUD	--	1A
	SQLDET	--	1E
	SQLVAL	--	28
	VOLMIN	--	26
	VOLVAL	A7	27
MODE			
	ACTMDU	--	1F
	ACTMDW	--	2F
	ACTSTU	--	49
	CHINFO	81	01 **(OBSOLETE)**
	CHLNUM	--	3F
	RPTDIR	--	24
	RXMODE	A1	21
	TRKMDU	--	46
	TXMODE	A0	20
	XTLPUL	--	54

FUNCTION	MNEMONC	RQST/BCST CODE (Hex)	
SPECIAL			
	ALARMS	--	47
	BATTST	--	3E
	CNFREQ	--	59
	DATCHN	--	4C
	DATOPC	--	4B
	DEVJSR	--	17
	EPREQ	--	06
	MEMACS	--	08
	MEMADD	87	07
	MEMFRA	83	03 **(OBSOLETE)**
	METINF	--	4F
	OPTSTS	96	16
	PRUPST	--	3B
	RADRDY	96	15
	SFTPT1	E6	56
	SGINFO	A5	25
	SPAOPC	--	5A
	SYSTAT	--	4D
	PRTVAL	--	44
	PWRCTL	--	48
	TESTOP	--	5B
	TIMEUP	--	4E
	TRKOPC	--	45
	TSTMOD	--	40
	VRSSTU	--	4A
	XFRBDY	--	51
	XFREND	--	52
	XFRERR	--	53

FUNCTION	MNEMONC	RQST/BCST CODE (Hex)	
DISPLAY/KEYBOARD			
	BUTCTL	--	57
	BUTPRS	--	14
	DELBUT	--	0F
	DECBUT	--	0C
	DEFBUT	--	55
	DISPLY	--	3C
	DSPMSG	--	3D
	ENTBUT	--	0D
	EXTBUT	--	0E
	INCBUT	--	0B
	LUMCTL	--	58
	RCLBUT	--	10
	REQBUT	--	11
	SETBUT	--	0A
	SHOBUT	--	09
CHANNEL SCAN			
	ACNPLB	--	38
	ACPRI1	--	39
	ACPRI2	--	3A
	ACTNPL	--	37
	ACTSTU	--	49
	SCONOF	--	2D
	TBONOF	--	31
	UNQSCN	--	30
CODED SQUELCH			
	ACRXPL	--	29
	ACTXPL	--	2A
	PLDECT	--	23
	REVINH	--	2C
	RXPLIN	--	2B
	TRKMDU	--	46

3.8.4 Numerical Opcode Table

Broadcast Code	Request Code	Mnemonic	Opcode Name
01	81	CHINFO	CHANNEL INFORMATION **(OBSOLETE)**
02	--	--	future framed opcode
03	83	MEMFRA	MEMORY ADDRESS/DATA FRAMED **(OBSOLETE)**
04	--	--	future framed opcode
05	--	--	future framed opcode
06	--	EPREQ	EXPANDED PROTOCOL REQUEST

IMPORTANT: The opcodes from $01 - $06 are allocated for what was formerly called 'framing'. Essentially, this means that any bus device that 'sees' one of these opcodes will suspend all bus activity until the SB9600 Busy line goes inactive. The device controlling Busy is allowed to keep it active for as long as it needs to, and normal SB9600 protocol timing does not apply. Any future use of these opcodes must be done carefully, taking the above issues into consideration.

Broadcast Code	Request Code	Mnemonic	Opcode Name
07	87	MEMADD	MEMORY ADDRESS/DATA
08	--	MEMACS	MEMORY ACCESS
09	--	SHOBUT	SHOW BUTTON
0A	--	SETBUT	SET BUTTON
0B	--	INCBUT	INCREMENT BUTTON
0C	--	DECBUT	DECREMENT BUTTON
0D	--	ENTBUT	ENTER CONFIGURATION
0E	--	EXTBUT	EXIT CONFIGURATION
0F	--	DELBUT	DELETE BUTTON
10	--	RCLBUT	RECALL BUTTON
11	--	REQBUT	REQUEST BUTTON
12	--	--	future button opcode

Broadcast Code	Request Code	Mnemonic	Opcode Name
13	--	--	future button opcode
14	--	BUTPRS	BUTTON PRESS
15	--	RADRDY	RADIO READY
16	96	OPTSTS	OPTION STATUS
17	--	DEVJSR	DEVICE SUBROUTINE JUMP
18	--	PTTINH	PTT INHIBIT
19	--	RADKEY	RADIO KEYED
1A	--	RXAUD	RECEIVE AUDIO ROUTING
1B	--	TXAUD	TRANSMIT AUDIO ROUTING
1C	--	ALRTTN	ALERT TONE
1D	--	AUDMUT	AUDIO MUTE
1E	--	SQLDET	SQUELCH DETECT
1F	--	ACTMDU	ACTIVE MODE UPDATE
20	A0	TXMODE	TRANSMIT MODE UPDATE
21	A1	RXMODE	RECEIVE MODE UPDATE
22	--	DISMUT	DISCRIMINATOR MUTE
23	--	PLDECT	PL DETECT
24	--	RPTDIR	REPEAT/DIRECT
25	A5	SGINFO	SIGNALLING INFORMATION
26	--	VOLMIN	VOLUME MINIMUM
27	A7	VOLVAL	VOLUME VALUE
28	--	SQLVAL	SQUELCH VALUE
29	--	ACRXPL	ACTIVE RECEIVE PL CODE
2A	--	ACTXPL	ACTIVE TRANSMIT PL CODE
2B	--	RXPLIN	RECEIVE PL MUTE CONTROL INHIBIT
2C	--	REVINH	REVERSE BURST INHIBIT
2D	--	SCONOF	SCAN ON/OFF
2E	--	TXCTRL	TRANSMIT CONTROL
2F	--	ACTMDW	ACTIVE MODE WRITE
30	--	UNQSCN	UNQUALIFY SCAN
31	--	TBONOF	TALKBACK ON/OFF
32	--	DEVVAL	DEVIATION VALUE
33	--	ACPRVL	ACTIVE POWER LEVEL
34	--	TXLTIN	TRANSMIT LIGHT

Broadcast Code	Request Code	Mnemonic	Opcode Name
35	--	ETONOF	EXTENDER ON/OFF
36	--	TOTVAL	TIME-OUT-TIMER VALUE
37	--	ACTNPL	ACTIVE NP LIST
38	--	ACNPLB	ACTIVE NP LIST BLANKING
39	--	ACPRI1	ACTIVE PRIORITY 1 MODE
3A	--	ACPRI2	ACTIVE PRIORITY 2 MODE
3B	--	PRUPST	POWER UP STATUS
3C	--	DISPLY	DISPLAY
3D	--	DSPMSG	DISPLAY MESSAGE
3E	--	BATTST	BATTERY STATUS
3F	--	CHLNUM	CHANNEL NUMBER
40	--	TSTMOD	TEST MODE
41	--	ACDADJ	ACTIVE DEVIATION ADJUST
42	--	PALIM	PA LIMIT VALUE
43	--	OSCVAL	OSCILLATOR VALUE
44	--	PRTVAL	PORT VALUE
45	--	TRKOPC	TRUNKING OPCODE
46	--	TRKMDU	TRUNKED MODE UPDATE
47	--	ALARMS	ALARMS
48	--	PWRCTL	POWER CONTROL **(OBSOLETE)**
49	--	ACTSTU	ACTIVE STATE UPDATE
4A	--	VRSSTU	VEHICULAR REPEATER STATUS UP
4B	--	DATOPC	DATA OPCODE
4C	--	DATCHN	DATA CHANNEL
4D	--	SYSTAT	SYSTEM STATUS
4E	--	TIMEUP	TIME UPDATE
4F	--	METINF	METROCOM INFORMATION
50	--	XFRSTR	TRANSFER START OF SEQUENCE
51	--	XFRBDY	TRANSFER BODY OF SEQUENCE
52	--	XFREND	TRANSFER END OF SEQUENCE

Broadcast Code	Request Code	Mnemonic	Opcode Name
53	--	XFRERR	TRANSFER SEQUENCE ERROR
54	--	XTLPUL	PROCESSOR CRYSTAL PULL
55	--	DEFBUT	DEFINE BUTTON
56	E6	SFTPT1	SOFT POT REGISTER #1
57	--	BUTCTL	BUTTON CONTROL
58	--	LUMCTL	ILLUMINATION CONTROL
59	--	CNFREQ	CONFIGURATION REQUEST
5A	--	SPAOPC	SPECIAL APPLICATIONS OPCODE
5B	--	TESTOP	TEST OPCODE

3.8.5 Opcode Descriptions

The following pages contain detailed information on each opcode. It should be noted that, unless otherwise mentioned, the transmitter of a message should send its own GROUP/ ADDRESS in the first packet. Also, B/R (bit 7 of the opcode) usually equals 0 (Broadcast), unless otherwise stated.

The vast majority of opcodes implemented in this system are related to radio or internal radio option controls. A cursory understanding of the suggested radio RAM structure is helpful in using these opcodes. The radio controls functions based on the Active RAM value (see Figure). This value can be modified directly by certain opcodes described in this section. Many opcodes which allow options to change radio parameters have two inhibit bits, D1 and D2. When an option sets D1 it inhibits radio changes to the associated parameter until D1 is cleared at system reset. In a similar manner, when an option sets D2, radio changes are inhibited until D2 is cleared by either Active Mode changes, transmitter de-key, (whichever is appropriate for that parameter) or system reset. The intent of the D2 bit is to relieve options from having to rewrite the old parameter when they are done with their function. When an option sets either bit, the operator is inhibited from changing the associated parameter (eg. a Selected Mode change will cause an Active Mode change unless D1 or D2 have previously been set). The radio ignores the D1 and D2 bits in Request opcodes. When both D1 and D2 are cleared the radio updates the parameter in its normal manner by writing Operator Selected or EEPROM values to it (internal inhibit and select bits determine which). Options which share a resource should remember the status of D1 and D2 to see if the resource is currently busy or not.This provides a crude first-come, first-served priority.

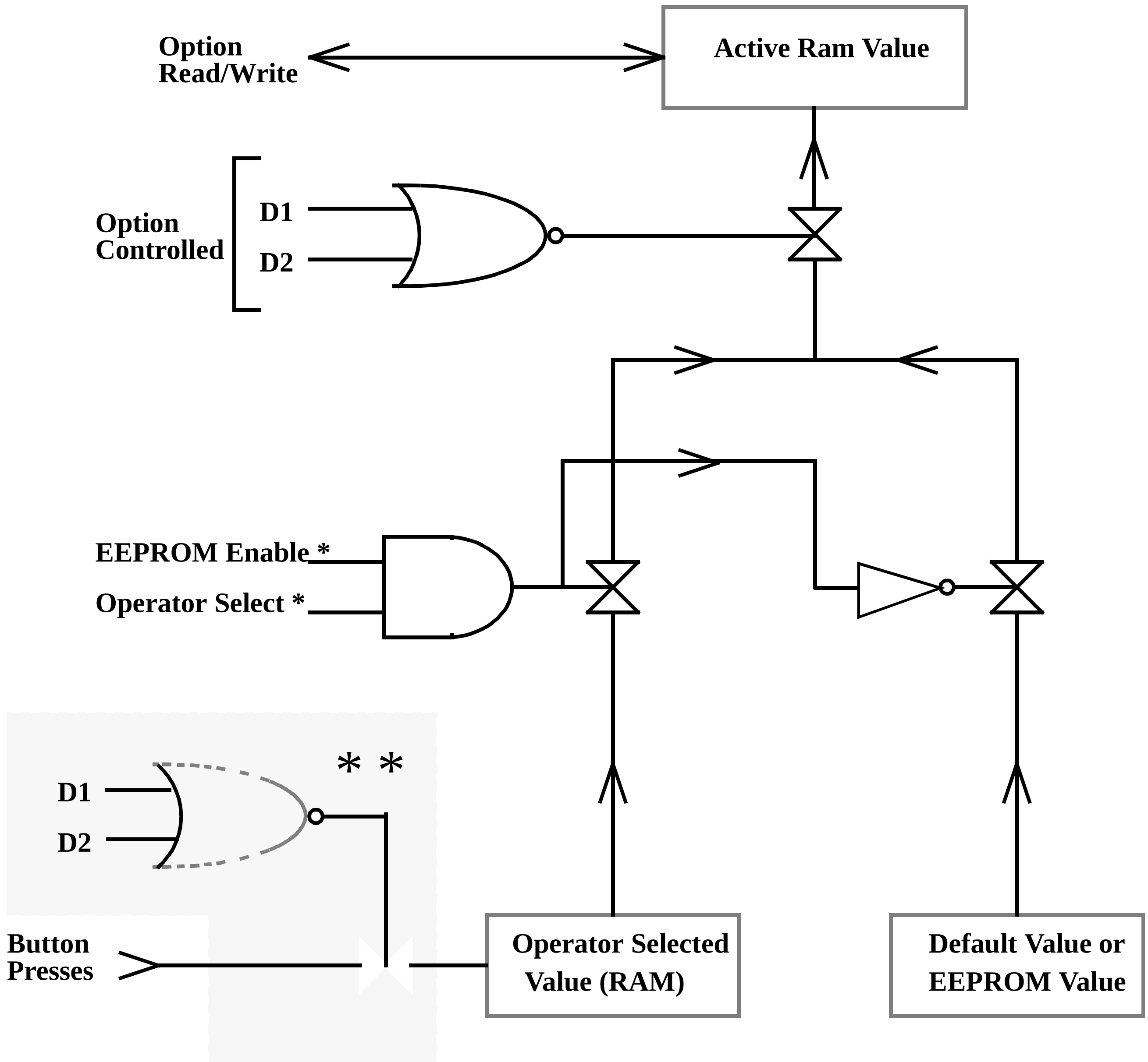
For unambiguous operation, the radio should remember operator selected parameters even while an option has re-written a new value. Otherwise, when the option releases the parameter, it will get reset to some fixed value. From the operators viewpoint, he has to keep resetting the value to his desired state after some event happens. Although not a requirement to fulfill complete compatibility with SB9600 such operation is considered undesirable.

Options may return the parameter to normal radio control by clearing D1 and D2. The parameter data is such a Broadcast is not used. The effect of Active Mode changes and transmitter de-keying on the state of the radio RAM bits controlled by these opcodes is indicated in the following section.

”PL” as used in this document refers to any continuously sent coded squelch including both tone and digitally coded squelch. ”MODE” refers to the RF channel and may include frequency, PL code, and other qualifying characteristics. ”Control head” is used to refer to the device that processes buttons, displays, and other human interfacing. If these processes are not handled by a single device, then only portions of the references to a ”control head” will be pertinent to each device.

I**MPORTANT: In all of the opcode descriptions that follow, if a field is described as ‘X’ or as an unused field, then that location must be written to ‘0’ in order to be forwards compatible with future definitions of those fields!**

l



* Sometimes implementedas a ”0” or ”1” in ROM

* * Boxed in area implemented on ON/OFF selects (e.g. scan ON/OFF and not implemented on value selections (e.g.scan list select)

Figure 3-2 Logical Representation Of (Option Controllable) Ram Structure

ACDADJ - ACTIVE DEVIATION ADJUST - opcode $41

This opcode is used by the options to alter the Active Deviation. This opcode is executed immediately upon receipt, regardless of the state of RADKEY.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active Deviation (DATA 3) updated only on system reset.

D2: Set this bit to have the Active Deviation (DATA 3) updated on transmitter de-key (when radio RF power is turned off), Active Mode changes, or system reset.

DATA 3 - 7 bit, signed value (2’s complement) to be added or subtracted from the default value.
= $01 to increment by 1 count
= $FF to decrement by 1 count

Either D1 or D2 must be set when an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active Deviation to the radio EEPROM selected value. Each ’count’ incremented or decremented represents a fixed dB change in deviation from the default value. Since different radios will may have different deviation sensitivities, the actual value an option sends for DATA 3 should be stored in its Option Status block.

ACNPLB - ACTIVE NP LIST BLANKING - opcode $38

This opcode allows options to delete all Non-Priority channels from the Active Non-Priority Channel Scan List with a single Broadcast. This opcode will typically be used with ACTNPL to quickly set up a new Active NP Scan List. This opcode should not affect the contents of the operator selected scan list.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

D1: Set this bit to have the Active NP list updated only on system reset.

D2: Set this bit to have the Active NP list updated on Active Mode changes or system reset.

When using this opcode in conjunction with ACTNPL, both opcodes must use the same D1/ D2 pair.

ACPRI1 - ACTIVE PRI 1 MODE - opcode $39

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 10/30/93 WITH VERSION 5.12

This opcode is used to change the Active Priority 1 Mode that is being used during Channel Scan. This opcode should not affect the operator selected value nor should it inhibit an operator from selecting an external Priority 1 Mode (ie. the radio should remember both the option defined and operator selected first priority). It should be noted that this opcode operates in the same fashion as ACPRI2. This opcode can be used whether scan is currently on or off.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active Pri 1 Mode (DATA 3) updated only on system reset.

D2: Set this bit to have the Active Pri 1 Mode (DATA 3) updated on Active Mode changes or system reset.

DATA 3: Contains the mode number of the Priority One mode.
- = 0, there is no Priority 1 Mode
- = 1, the Priority 1 Mode is ”Mode 1”
- = 2, the Priority 1 Mode is ”Mode 2”
- etc.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active Priority 1 Mode to the radio EEPROM or operator selected value.

ACPRI2 - ACTIVE PRI 2 MODE - opcode $3A

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 10/30/93 WITH VERSION 5.12

This opcode is used to change the Active Priority 2 Mode that is being used during Channel Scan. This opcode should not affect the operator selected value nor should it inhibit an operator from selecting an external Priority 2 Mode (ie. the radio should remember both the option defined and the operator selected second priority). It should be noted that this opcode operates in the same fashion as ACPRI1. This opcode can be used whether scan is currently on or off.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active Pri 2 Mode (DATA 3) updated only on system reset.

D2: Set this bit to have the Active Pri 2 Mode (DATA 3) updated on Active Mode changes or system reset.

DATA 3: Contains the mode number of the Priority Two mode.
= 0, there is no Priority 2 Mode
= 1, the Priority 2 Mode is ”Mode 1”
= 2, the Priority 2 Mode is ”Mode 2”
etc.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active Priority 2 Mode to the radio EEPROM or operator selected value.

ACPRVL - ACTIVE TX PWR VALUE - opcode $33

This opcode is used by options to change the Active RF Transmitter Power Output level. This opcode should not affect the operator selected value nor should it inhibit an operator from selecting an external Transmit Power Level (ie. the radio should remember both the option defined level and the operator selected level). Since different radios have different numbers of bits to represent RF power and each increment is a different power change, general purpose options should store the desired value in their Option Status block. This opcode is executed immediately upon receipt, regardless of the current state of RADKEY.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active TX Pwr Value (DATA 3) updated only on system reset.

D2: Set this bit to have the Active TX Pwr Value (DATA 3) updated on transmitter de-key (when radio RF power is turned off), Active Mode changes, or system reset.

DATA 3: Contains the Active Power Value
= $00 for lowest power
= $FF for highest power

Either D1 or D2 must be set when an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active Power Value to the operator or radio EEPROM selected value.

ACRXPL - ACTIVE RECEIVE PL CODE - opcode $29

This opcode allows options to change the Active PL code used during receive. This opcode should not affect the operator selected value nor should it inhibit an operator from selecting an external Receive PL Code (ie. the radio should remember both the option defined code and the operator selected code).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active Receive PL Code (DATA 3) updated only on system reset.

D2: Set this bit to have the Active Receive PL Code (DATA 3) updated on Active Mode changes or system reset.

DATA 3: Contains the PL code number. This number refers to a list of codes stored in the radio EEPROM. If DATA 3 = 0, the PL code is Carrier Squelch.

Either D1 or D2 must be set when an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active PL Code to the operator or radio EEPROM selected value.

ACTMDU - ACTIVE MODE UPDATE - opcode $1F

This opcode is broadcast by the radio whenever the Active mode changes to a conventional non-trunked mode due to an operator Button Press. It is also sent in response to an ACTMDW opcode. It contains the Option Enables for the Active Mode. Options cannot change modes by Broadcasting this opcode, they must use ACTMDW. If the Active mode is a trunked mode, then TRKMDU will be sent instead of ACTMDU.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1 - Contains the Option Enables which relate to the Active Mode. The Option Enables are one byte of radio EEPROM information for each mode which are used by options for enabling certain option features on a mode- by-mode basis. Options interpret these bits based on their Option Status bytes. Each option can assume that its bits are contiguous but relocatable (the Option Status register must indicate which bits to use).

DATA 2 - contains the Active Mode Number

If an option is only valid for trunked systems, then it may turn itself off when this opcode is received. Its Option Enables may then be used by another option.

ACTMDW - ACTIVE MODE WRITE - opcode $2F

This opcode is broadcast by options to change the Active Mode. It should not inhibit an operator from selecting another mode nor should the opcode affect the operator selected mode (ie. the radio should remember both the option defined mode as well as the operator selected mode). The radio will respond to this opcode with an ACTMDU indicating what the new mode and Option Enables are, even if the actual mode of operation doesn't change. This opcode can be used for data steering. It should be noted that to monitor on the data steered channel, PTT and scan must be inhibited on that mode. Also, care must be taken when using ACTMDW with Talkback scan - the radio may return to the operator selected mode and not the Talkback Mode when D1 and D2 (described below) are cleared to zero. This is the only opcode that prohibits a Selected Mode change from causing an Active Mode change.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
Data 3							
0	Opcode						
CRC							

D1 - Set this bit to have DATA 3 (Active Mode) updated only on system reset.

D2 - Set this bit to have DATA 3 (Active Mode) updated on transmitter de-key or system reset.

DATA 3 - Contains the Active Mode Number.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Active Mode is updated to the operator selected value. If DATA 3 equals zero or is an invalid channel number, then the mode will not change, although D1 and D2 will still be acted on. Options utilizing this opcode may want to monitor ACTMDU to make sure the radio reached the desired channel before proceeding with their normal operation. This will also allow options which use ACTMDW to determine when another option has usurped their control. In this case, options should suspend or abort their operation until the mode has been restored (ACTMDW with D1=D2=0). Options may also be designed to use ACTMDW only if it is currently "available". Currently there are no priorities associated with ACTMDW.

ACTNPL - ACTIVE NP LIST - opcode $37

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 10/30/93 WITH VERSION 5.12

This opcode allows options to modify the Active Non-Priority Channel Scan List. This opcode should not inhibit an operator from configuring an external NP scan list nor should the opcode affect the operator selected NP scan list (ie. the radio should remember both the option defined scan list as well as the operator selected scan list).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
D1	D2	x	Data 3				
Data 4							
0	Opcode						
CRC							

D1: Set this bit to have the Active NP list updated only on system reset.

D2: Set this bit to have the Active NP list updated on Active Mode changes or system reset.

DATA 3: This data represents which block of the NP list is being accessed
= 0, modes 1 thru 8
= 1, modes 9 thru 16
etc.

DATA 4 - This data indicates whether each mode in the block pointed to by DATA 3 is on the NP list or not. Bit 7 of DATA 4 corresponds to the lowest number mode. A ‘1’ means the mode is on the list, a ‘0’ means the mode is not on the list.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active Non-Priority list to the radio EEPROM or operator selected value. When using this opcode in conjunction with ACNPLB, both opcodes must use the same D1/D2 pair.

ACTSTU - ACTIVE STATE UPDATE - opcode $49

This opcode is broadcast by the radio whenever any of its miscellaneous On/Off controls change state. These can change due to mode-slaving, operator Button Presses, or by receiving RPTDIR, SCONOF, ETONOF, or TBONOF opcodes. These opcodes are all used by options to attempt to change the state of the radio. ACTSTU is to be used to monitor whether or not the change was successful (eg. perhaps another feature or a configuration state inhibited a function from being turned on).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Reserved but currently undefined

DATA 2: Contains the state of various On/Off radio controls as shown below:
- bit 7 - Scan On/Off (1=On)
- bit 6 - Talkback On/Off (1=On)
- bit 5 - reserved
- bit 4 - RPT/DIR (1=DIR)
- bit 3 - Extender On/Off (1=On)
- bit 2 - reserved
- bit 1 - reserved
- bit 0 - reserved

ACTXPL - ACTIVE TRANSMIT PL CODE - opcode $2A

This opcode allows options to change the Active PL code used during transmit. This opcode should not affect the operator selected value nor should it inhibit an operator from selecting an external Transmit PL Code (ie. the radio should remember both the option defined code and the operator selected code).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
Data 3							
0	Opcode						
CRC							

D1: Set this bit to have the Active Transmit PL Code (DATA 3) updated only on system reset.

D2: Set this bit to have the Active Transmit PL Code (DATA 3) updated on transmitter de-key (when RF power is turned off), Active Mode changes, or system reset.

DATA 3: Contains the PL code number. This number refers to a list of codes stored in the radio EEPROM. If DATA 3 = 0, the PL code is Carrier Squelch.

Either D1 or D2 must be set when an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio will update the Active PL Code to the operator or radio EEPROM selected value.

ALARMS - ALARMS - opcode $47

This opcode is Broadcast by the radio whenever the selected Alarms have changed. It is to be used to help multiple options access the alarms in a radio system. Although each option may have its own alarm display handling, in a multiple option installation only one option should handle the alarm display and alarm selection process. Alternatively, the radio can handle the alarm displays and indicate to the options which alarms are selected by Broadcasting this opcode. Only one device in an installation is allowed to handle alarm displays and Broadcast this opcode. Also, this opcode may be sent during a power-up sequence if any of the alarms are selected on turn-on.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	D2	D1
0	Opcode						
CRC							

D1: = 1 if the Horn alarm has been selected

D2: = 1 if the Lights alarm has been selected

Note that Group = 0 allowing every device to receive this opcode. Also note that DEVICE ADDRESS will almost always be $01 (unless some other device is handling the alarm displays). More alarms may be added in the future by expanding the second data byte.

ALRTTN - ALERT TONE - opcode $1C

This opcode is used by options and the control head to control the radio-generated alert tones. The radio will automatically adjust to minimum volume if necessary (see VOLMIN opcode for details of this feature). The radio has priority over all alert tone use when using the alert tone for internal functions such as TOT. The alert tone can be used regardless of the state of RXAUD (audio is automatically muted).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
D1	Data 2		Data 3				
Data 4							
0	Opcode						
CRC							

D1: Set this bit to have the radio continuously repeat the pattern (DATA 4)

DATA 2: = PRIORITY = 00 for lowest priority
= 01 for low priority
= 10 for high priority
= 11 for highest priority

DATA 3: = FREQUENCY in 100 Hz increments
= 0 for 000 Hz
= 1 for 100Hz
.
.
= 31 for 3100 Hz

DATA 4: Contains the Pattern Number described in Appendix A

When a device wishes to send a continuous pattern, it sets D1=1. Other options are then not allowed to use ALRTTN unless they have a higher priority continuous pattern (the radio might ignore lower priority patterns). When a device wishes to end a continuous pattern, it must send ALRTTN with D1=0 AND DATA 4 = 0. Other options are then free to send ALRTTN for their own use. Options must remember the last state of D1 for them to know if the alert tone is available (D1=0) or not (D1=1). When a single pattern sequence is desired, ALRTTN is sent with D1=0. When several ALRTTN opcodes are sent (D1=0),

the radio will generate the pattern with the highest priority. All others will be ignored. Note that because of D1, all continuous patterns have a higher priority than any single pattern. At system reset, D1 and DATA 4 are reset to 0.

AUDMUT - AUDIO MUTE - opcode $1D

This opcode is Broadcast by the radio to inform options when the audio mute line, which controls speaker muting, changes state. It is used to indicate when the speaker of the radio has audio present. Typical uses include informing a rear control head or external speaker when to mute/unmute its audio to conform with audio heard at the normal speaker.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	D1
0	Opcode						
CRC							

D1: = 0 audio is muted
= 1 audio is unmuted

BATTST - BATTERY STATUS - opcode $3E

This opcode is Broadcast on power up after RADRDY to indicate the quantity of charge remaining in the battery.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Relative charge remaining

DATA 2: Relative voltage remaining

These quantities are relative to the initial value of 255. The charge is calculated based on a time/voltage relationship specific for each particular battery.

These quantities are relative to the initial value of 255. The charge is calculated based on a time/voltage relationship specific for each particular battery.

BUTCTL - BUTTON CONTROL - opcode $57

This opcode is used to indicate the press and release of physical and ‘virtual’ buttons and the position of various switches.

Normally, GROUP and DEVICE ADDRESS are those of the source ($05 for control head #1 and $06 for control head #2, for example), but refer to the discussion about virtual button presses for further details.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

Refer to the following pages for the specific definitions of Data1 and Data2.

BUTCTL Button Control $57

This opcode provides a means to "manipulate" both the physical and virtual buttons of a radio. Its usage allows an option to tell the radio to behave as if the designated status change of a button had occurred.

For the purpose of this opcode description, a button is defined to be any non-RF input source to the radio. The term "button" includes physical push-and-release buttons, push-and-hold buttons, switches, rotaries, vehicle interface ports (VIPs), and the virtual counter-parts of each.

The intent is that this opcode will be used by, but not limited to, automated test systems and control heads. Automated test systems will typically confine themselves to the usage of virtual buttons. While control heads are absolutely confined to the usage of physical buttons. The structural difference in this opcode between physical and virtual buttons is the DEVICE ADDRESS value of the message.

The radio will broadcast this opcode during Control Head Test Mode when the status of a button or switch has changed states.

SB9600 DEFINITION
Byte 1: GROUP / DEVICE ADDRESS
Byte 2: DATA1
Byte 3: DATA2
Byte 4: OPCODE
Byte 5: CRC

GROUP /DEVICE ADDRESS:
bits 7-5: GROUP
bits 4-0: DEVICE ADDRESS

DEVICE ADDRESS Usages:
$01 => Virtual Button
$05 => Physical Button from a Control Head
$06 => Physical Button from a Control Head

DATA1:
bit 7: Register Value(msb)
bit 6: Register Value
bit 5: Register Value
bit 4: Register Value
bit 3: Register Value
bit 2: Register Value
bit 1: Register Value
bit 0: Register Value (lsb)

Register Value:
Contains the Button ID value representing the desired physical or virtual button.
$00-$FD Valid button registers
$FE Control Register for TAKING CONTROL.

See DATA2 description for more details.

$FF Control Register for RELEASING CONTROL.
See DATA2 description for more details.

DATA2:
There are three types of buttons:

- Momentary, which are either released or pressed
- Static, which can take any of several values
- Free Revolving Rotaries

In addition, there is the ability to TAKE and RELEASE CONTROL of a button as defined by DATA1. The Button register mapping dictates whether a physical button is momentary, static, or free revolving.

DATA2 TAKE CONTROL (DATA1=$FE)

Contains the Button Register of the button over which control is being taken. When control is TAKEN, the radio ignores any changes that it perceives to the indicated button; the changes may be sourced from either the internal physical counterpart or from the serial bus. ONLY changes coming from the source that has taken control will be effective.

$00-FD Take Control of the Button Register indicated in DATA2
$FE Take Control of all Momentary buttons
$FF Take Control of all Static buttons

DATA2 RELEASE CONTROL (DATA1=$FF)

Contains the Button Register of the button over which control is being released. When Control is released, the radio will again process any changes that it perceives to the indicated button; the changes may be sourced from either its own internal physical counterpart or from the serial bus.

$00-FD Release Control of the Button Register indicated in DATA2
$FE Release Control of all Momentary buttons
$FF Release Control of all Static buttons
DATA2 Momentary Button:
$00 Release Button
$01 Press Button
$02 Press & Immediately Release Button
$03 - $FF reserved for future expansion

DATA2 Static Button:
Contains the discrete value for the static button.
Each static button has its own range of valid values; a range of 0 => 255 is available.
Radio platforms will ignore attempts to set a static button to an unsupported position.

DATA2 Free Revolving Rotary:
This is a signed value representing a relative delta from the last time the position of this input was reported. The sign will indicate whether the move was a clockwise or a counter-clockwise movement. Clockwise movement will be indicated by a positive value.

Usage Notes:

DEVICE ADDRESS:
The DEVICE ADDRESS of a radio is $01. Whenever a radio receives its own DEVICE ADDRESS in the BUTCTL opcode, the button register in DATA1 is to be interpreted as a VIRTUAL BUTTON.

The DEVICE ADDRESSes of control heads are $05 and $06. Whenever a radio receives the DEVICE ADDRESS of a control head in the BUTCTL opcode, the button register in DATA1 is to be interpreted as a PHYSICAL BUTTON.

PHYSICAL vs VIRTUAL BUTTONS:
All radios will physically be equipped with user input buttons. However, the actual number of buttons physically residing on a radio will vary based on radio platform and radio model. The buttons that ***physically*** reside on a radio are referred to in this opcode description as PHYSICAL BUTTONS.

The software within the radio may be capable of supporting more buttons than physically reside on the radio. That is, if the button COULD be pressed, the software would be capable of responding to it. The buttons that the radio is ***capable*** of recognizing are referred to in this opcode description as VIRTUAL BUTTONS. Note that this means that the possibility exists to define Virtual Buttons that are not implemented physically.

PHYSICAL BUTTONS are a subset of the buttons that are supported by a software package. The VIRTUAL BUTTONS represent ALL of the buttons that the software is capable of supporting.

VIRTUAL BUTTONS can be used to provide a standard interface to a feature. The PHYSICAL BUTTONS then become the set of buttons needed to give the User access to that feature or its interface. If another processor is the User of the radio via the SB9600 serial bus, BUTCTL's access to the VIRTUAL BUTTONS allows the processor to be independent of radio platform and model number limitations. Test Mode takes advantage of this opportunity to provide a Standard Test Mode Interface.

Since Feature Interfaces are, by nature, functionally specific, the use of the VIRTUAL BUTTON registers becomes VERY functionally oriented rather than purely physical in nature. For example, if the radio is in STATE-A and it receives a BUTCTL message indicating that Control Head $05 detected that the physical button with register $19 was pressed, the radio would be required to determine what that button means in STATE-A, before acting on the button press. However, if the radio, still in STATE-A, were to receive a BUTCTL message which had the radio's DEVICE ADDRESS and button register $97 and indicating that the button was just pressed, the radio would already know what to do and would require no button-to-state validation. In addition, the two messages discussed could have the SAME effect on the radio's operation. The difference would be that the Virtual Button $97 is part of the defined interface while the physical button $19 is a nonstandard means of reaching the feature.

Since the assignment of VIRTUAL BUTTON registers to radio-state functionality is arbitrary except for making sure no ambiguities exist, the Standard Test Mode Interface has assigned the VIRTUAL BUTTON registers identically to what the PHYSICAL BUTTON registers would be if the buttons were supported. Notice that this prevents ambiguity while allowing the radio to ignore the DEVICE ADDRESS and allows the User the same access to Test Mode that an option has, provided that all required buttons physically exist on the radio. Note that the BUTCTL opcode IS NOT intended to be used by options other than control heads. Test Mode, however, is anticipated to be interfaced to a Factory Test System or a Radio Service Software platform. In the future, should there be any ambiguity or incompatibility between the virtual and physical buttons, the radio will test the DEVICE ADDRESS. All senders of this opcode MUST use the correct DEVICE ADDRESS now.

Note that it is possible for a Virtual Button to have more than one interpretation within a radio, but that for any one state of the radio, there is only one interpretation. For example, Radio State A may provide a virtual interface that associates Function Z to Virtual Button P and Radio State B may provide a virtual interface that associates Function Y to Virtual Button P. The same virtual button is used by different radio states, but for different purposes.

There is a standard for Button Register Mapping that is used to assign Button IDs(used in DATA1) to buttons and is described in the appendix A section called BUTCTL Register Assignment Standard. ALL FUTURE RADIOS AND CONTROL HEADS SHOULD ADHERE TO THIS STANDARD.

For a description of the Virtual Interface Definition for Test Mode, refer to the Software Requirements Specification section of Test Mode (Document Control Number: FL08-RQMT-91A009).

Examples:

MOMENTARY BUTTON: Press Internal PTT
DATA1: $01
DATA2: $01

MOMENTARY BUTTON: Release Internal PTT
DATA1: $01
DATA2: $00

MOMENTARY BUTTON: Quick Press First Front Function Button
DATA1: $80
DATA2: $02

TAKE CONTROL: Take Control of the Volume Continuous Rotary
DATA1: $FE
DATA2: $02

RELEASE CONTROL: Release Control of the Volume Continuous Rotary
DATA1: $FF
DATA2: $02

TAKE CONTROL: Take Control of all Momentary Buttons
DATA1: $FE
DATA2: $FE

RELEASE CONTROL: Release Control of all Momentary Buttons
DATA1: $FF
DATA2: $FE

DISCRETE BUTTON: Select Position 3 of the first Rotary with less than 4 positions
DATA1: $40
DATA2: $03

BUTPRS - BUTTON PRESS - opcode $14

This opcode is used by a control head to inform other devices that one of the buttons has been pressed. This opcode is used when the operator needs to input a sequence of numbers (eg. a telephone number) which is edited by the option, as opposed to a locally edited number ranging from 0 to 255. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular device

DATA 2: Contains the ASCII representation of the button pressed on the control head. Examples include character delete ($7F), and the normal alphanumeric characters.

Both GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

CHINFO - CHANNEL INFORMATION - opcode $01

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 12/9/92 WITH VERSION 5.1

This Framed opcode is used to read or write up to a 16 byte block of information associated with a specified RF channel. Such information is radio specific and can contain different information from radio to radio. Thus only programmers or devices with intimate knowledge of the destination device software structure should use this opcode. See Section (none) regarding Framed opcodes and Section 3.3 regarding Request opcodes.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
D1	x	x	x	x	x	x	D2
Data 3							
B/R	Opcode						
CRC							

D1: = 0 to unlock Radio Channel Switch
= 1 to lock Radio Channel Switch

D2: = 0 to transfer all of the channel data
= 1 to transfer only the Receive data of the channel

DATA 3: Contains the channel number

To change the channel information, a device sends a Broadcast CHINFO (B/R=0) with the desired channel number and data type. This is then followed by the appropriate number of bytes of data. To read channel information, a device sends a Request CHINFO (B/R=1) setting the desired channel number and data type. The destination device then sends the Broadcast CHINFO (B/R=0) with the specified channel number and data type followed by the appropriate number of bytes of data.

CHLNUM - CHANNEL NUMBER - opcode $3F

This opcode is broadcast by an option to change the TX/RX frequency which the existing mode uses. Neither the Active mode number, nor the Selected mode number changes, although the actual RF frequency does. Typical uses are for ”System Access” type options (eg. trunking) where the option doesn’t want to change modes (eg. trunking is active only on the existing mode) but does want to change the actual frequency of operation (via a command from the trunking control channel).

Bit

7	6	5	4	3	2	1	0
Group			D1	x	x	D2	D3
Data 1							
Data 2							
0	Opcode						
CRC							

D1: = 1 to indicate 800Mhz band, 0 to indicate some other band

D2: = 1 to indicate channel spacings based on 6.25KHz
= 0 to indicate channel spacings based on 5KHz

D3: = 1 for receive, 0 for transmit

DATA 1: Contains the MSB of the Channel Number

DATA 2: Contains the LSB of the Channel Number

It should be immediately noted that the DEVICE ADDRESS field has been redefined to accommodate more data. Since this opcode is normally sent by the trunking option, this should present no difficulty in determining who sent this opcode.

The frequency specified by this opcode remains valid until:

1) CHLNUM is again sent with DATA1=DATA2=$00, in which case the normal mode definitions are restored,
2) Active mode changes, normally due to ACTMDW and Button opcodes, or
3) System Reset. In other words, it is treated as if the ”D1-D2 inhibit bits” are set such that D1=0 and D2=1.

DATCHN - DATA CHANNEL - opcode $4C

This opcode is used to build a data channel frequency list in the radio. The data option has the capability to steer the radio to select the new data upon subsequent data channel requests. This opcode does not make a request or cause the radio to synthesize a new data channel; it simply adds to the data channel list in the radio. In order to get a new data channel or change the current one, the data option must still send a data channel request or a change data channel opcode. These data channels may be purged from the radio's list if they become inactive after the radio receives OSW grants for them.

Bit

7	6	5	4	3	2	1	0
Group			D1	D2	0	0	D3
0	0	0	0	0	0	Data 1	
Data 2							
0	Opcode						
CRC							

D1: = 0; Other Band channel number format
= 1; 800 Mhz channel number format

D2: = 0; Add data channel to radio list without steering to it
= 1; Add data channel to radio and steer to this channel

D3: = 0; UHF Tx channel number
= 1; 800 or UHF Rx channel number

DATA 1: = 2 most significant bits of 10 bit channel number.

DATA 2: = 8 least significant bits of 10 bit channel number.

For Other Band, the Rx and Tx channel numbers are always sent Rx followed by Tx channel number. If two consecutive Rx channel numbers are sent, the first channel number will be lost. If two consecutive Tx channel numbers are sent, the last Tx channel number will be lost. The format of the 10 bit channel number is the same as that used in the Smartnet Type II Protocol Specification document (0 - 759).

Upon serial bus reset, ACTMDU, or TRKMDU opcodes, the radio and options should erase any record of data channels added by this opcode.

DATOPC - DATA OPCODES - opcode $4B

This opcode is Broadcast by the radio or a data option in order to implement the special functions required by using the data option on a trunking channel. This opcode is used in conjunction with DATCHN and SYSTAT. This is one of the few opcodes that doesn't use the ADDRESS bits.

Bit

7	6	5	4	3	2	1	0
Group			Data 1				
Data 2							
Data 3							
0	Opcode						
CRC							

DATA 1: Contains data associated with the data opcode

DATA 2: Contains data associated with the data opcode

DATA 3: Contains the special data opcode

See APPENDIX H for a description of the data option sub-opcodes.

DECBUT - DECREMENT BUTTON - opcode $0C

This opcode is Broadcast by control heads to indicate that a scrolling type button has been pressed, in particular a scroll down button. See Section 3.5.1 for a discussion on button handling.

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Group			Device Address				
Data 1							
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

DEFBUT - DEFINE BUTTON - opcode $55

This opcode is broadcast by an option to change the definition of the button that is currently being pressed. It is sent in response to a button opcode sent by a control head. Upon receiving this opcode, the control head will redirect the button to the device specified in GROUP/ADDRESS and redefine the button according to the data specified.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
New Button Register							
New Button Opcode							
0	Opcode						
CRC							

GROUP/ADDRESS: Contains the redefinition for the current button destination group/ address

NEW BUTTON REGISTER: Contains the redefinition for the current button register

NEW BUTTON OPCODE: Contains the redefinition for the current button opcode

These redefinitions remain valid until the user lets go of the current button in which case these redefinitions are completely forgotten (ie., if the user presses the button again, this opcode must be sent again). This opcode will be ignored if the previous button pressed (and broadcast on the bus) is not the same as the button being pressed when the opcode is received.

A typical use for this opcode is to prevent a non-programmable control head from executing an ENTBUT when a button is held, but instead execute an INCBUT. Converting an INCBUT to an ENTBUT is also possible. It should be noted that there is no provision to redefine button data. Button data sent in the subsequent redefined button broadcast may be meaningless.

DELBUT - DELETE BUTTON - opcode $0F

This opcode is Broadcast by control heads to indicate that the user wishes to edit a selection. It can be used to alter a list (eg. operator select scan) or modify a numeric entry (DTMF), for example. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

DATA 2: Contains any data associated with the button

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

Each channel of a band (Lowband, VHF, UHF, 800, and 900) has a number assigned to it. For example, for 800MHz Motorola trunking, Channel 1 corresponds to 851.0125MHz (mobile Rx) while Channel 199 corresponds to 855.9875MHz (mobile RX). Definitions of channel number assignments for each bands can be found in the appropriate document describing the system utilizing them. In particular, the UHF band uses some of the Device Address field as DATA 3, and so care must be taken with this opcode.

DEVJSR - JUMP - opcode $17

This opcode allows a programmer or option to cause a device to execute a subroutine jump. This may be used to execute test routines that have been loaded into RAM or EEPROM or to execute SP subroutines located in EEPROM. This opcode is radio specific and any device using it must be intimately familiar with the specified device. If used during realtime operation, special care must be taken to not violate any timing constraints imposed by the internal software structure of the specified device (ie. the subroutine can't be too long or it won't work). The specified address is absolute - up to 64k can be addressed.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1 (MSB) : DATA 2 (LSB): = The absolute address of the subroutine to be executed.

The GROUP and DEVICE ADDRESS specifies the destination device instead of the source device (the normal case).

DEVVAL - DEVIATION VALUE - opcode $32

This opcode and its associated values are used to set the transmitter deviation. Since different radios have different numbers of bits to represent deviation and each increment is a different deviation change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure. General purpose options wishing to modify their deviation settings should use ACDADJ.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the reference oscillator deviation setting

DATA 2: Contains the VCO deviation setting

Both DATA 1 and DATA 2 are restored to the EEPROM value on 1) Active Mode changes, or 2) system reset. In other words, it is treated as if the ”D1-D2 inhibit bits” are set such that D1=0 and D2=1.

DISMUT - DISCRIMINATOR MUTE - opcode $22

This opcode is used by options to control muting of discriminator audio. This mute control can be considered a more 'permanent' mute than that obtained by RXAUD (which can also mute the discriminator). Options desiring a 'long term' mute (eg. Selective Call) or mute control without affecting RXAUD (eg. Data Operated Squelch - DOS) are advised to use this opcode to allow normal use of RXAUD. Also, any option which processes discriminator audio and places the result on RXAUD (eg. DVP) should also mute their audio when this opcode is in effect. Thus, this opcode can be used to mute discriminator audio, regardless of whether it comes straight from the radio or after being processed by other options.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	D1
0	Opcode						
CRC							

D1: = 0 to not affect the discriminator output.
=1 to disconnect the discriminator output from the audio path.

The 'normal' state is D1=0. If any option writes D1=0 and a second option still desires muting, it must again broadcast DISMUT with D1=1. It is recommended that options remember the last state of D1 in order to reduce bus activity due to redundant mute commands (ie. don't send DISMUT with D1=1 if D1 is already set). Also, to guarantee that it never gets 'stuck' muted, if an option mutes the discriminator, it should also release (unmute) it when done, as opposed to relying on another device to do it. D1 is cleared on system reset.

DISPLY - DISPLAY - opcode $3C

This opcode is used by the radio and options to write to the control head display or control system output ports. See Section 3.5.2 for a discussion on display handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the FIELD ID

DATA 2: Contains the DISPLAY DATA

The FIELD ID can be thought of as a control byte instructing the control head where to put the data. Fields are typically divided into functional areas such as the entire display, the TX indicator, the mode description (whether it be ASCII or mode number), SCAN ON, SCAN OFF, horn relay, etc. For displays that do not have control head defined data (eg. a telephone number), the data is included in the Display Data byte. Other examples of FIELDS and DISPLAY DATA can be found in supplements describing the individual devices which use the display. Each device will define its own set of FIELDS. A programmable display will require non-volatile storage to map these pre-determined fields into physical display locations.

GROUP/ADDRESS are sent as $00 when writing to the system FIELDS ($00 thru $0F).

DSPMSG - DISPLAY MESSAGE - opcode $3D

This opcode is used by devices to write a message to the control head display. A message is different than a normal display since it is ’latched’ at the control head and remains ’on’ until the operator performs some option-dependent operation. The option then ’releases’ the message (ie. the display goes back to its normal mode of operation). See Section 3.5.2 for a discussion on display handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the FIELD ID

DATA 2: Contains the DISPLAY DATA
= 0 to release the message

The FIELD ID can be thought of as a control byte instructing the control head where to put the data. Fields are typically divided into functional areas such as the entire display, the TX indicator, call light, etc. For displays that do not have control head defined data (eg. a call number), the data is included in the DISPLAY DATA byte. Other examples of message FIELDS and DISPLAY DATA can be found in supplements describing the individual devices which use the display. Each device will define its own set of FIELDS. To release a message display field, the FIELD should be sent with DATA = $00. A programmable display will require non-volatile storage to map these pre-determined fields into physical display locations.

ENTBUT - ENTER CONFIGURATION - opcode $0D

This opcode is Broadcast by control heads to indicate that the user wishes to modify the operation of an option. It can also be used when a shared keypad is in its numeric entry mode (as opposed to an option ON/OFF mode). Thus, before an option allows numeric entry via a keypad, it may be required to first enter into a configuration mode. The keypad is released when the EXTBUT is sent (see EXTBUT opcode). Options enter into a configuration mode for unit ID selection (selective calling) or key selection (multi-key DVP), for example. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

EPREQ - EXPANDED PROTOCOL REQUEST - opcode $06

This opcode is used by options (typically a system programmer) to instruct the device that the “Serial Bus Expanded Protocol” (SPEP), or some other future protocol perhaps, will subsequently be used. Use of this opcode halts normal SB9600 BUS activity. The Group in the first byte = 000 so that all devices will ‘hear’ it. The Device Address in the third byte can never be 00000 when changing protocols to SBEP since SBEP is designed to only be used between two devices, never more. Appendix I of this document provides details of SBEP.

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	0
SPARE		P-COL		BAUD			
Group			Device Address				
0	Opcode						
CRC							

SPARE: = Reserved, always 0

P-COL: = Protocol Selection

00	SB9600
01	SBEP
10	Future
11	Future

BAUD: = Baud rate for SBEP

0000	38400
0001	19200
0010	9600
0011	4800
0100	2400
0101	1200
0110	600
0111	300

Opcode $06

Upon receiving this opcode selecting SBEP as the new protocol, devices must check to see if the address matches theirs. If not, then they are prohibited from transmitting anything on the bus. The devices must monitor the busy line, and once it is released, they can return to normal operations.

ETONOF - EXTENDER ON/OFF - opcode $35

This opcode allows options to modify the state of the noise blanker in extender-equipped Low Band radios. It should not inhibit an operator from selecting another state nor should the opcode affect the operator selected state (ie. the radio should remember both the option defined state as well as the operator selected state). This opcode should only be used in an attempt to change the state of the extender. Use ACTSTU to monitor the current state of the extender.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
X	X	X	X	X	X	X	D3
0	Opcode						
CRC							

D1: Set this bit to have D3 updated only on system reset.

D2: Set this bit to have D3 updated on Active Mode changes or system reset.

D3: = 0 to turn the Extender off.
= 1 to turn the Extender on.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Extender is updated to the radio EEPROM or operator selected state.

EXTBUT - EXIT CONFIGURATION - opcode $0E

This opcode is Broadcast by control heads to indicate that an option is done being modified. It is also used when a shared keypad reverts to its normal option ON/OFF mode (as opposed to the numeric entry mode). The control head sends EXTBUT, allowing an option to exit the configuration mode, when the operator signifies that he is done with the option (such as hitting a HOME or SELECT key) or in response to an EXTBUT sent by an option. Any data associated with this opcode may indicate the latest value the operator has selected.

This opcode is also sent by options to cause the control head to exit configuration mode without requiring operator interaction (exception: see REQBUT). However, the configuration mode will not be exited until the control head responds with its own EXTBUT. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

DATA 2: Contains any data associated with the button

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case) when sent by the control head. When sent by an option, the source address is used.

INCBUT - INCREMENT BUTTON - opcode $0B

This opcode is Broadcast by control heads to indicate that a scrolling type button has been pressed, in particular a scroll up button. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

MEMACS - MEMORY ACCESS - opcode $08

This opcode is used by options (typically a system programmer) to read/write memory in the various locations in the system. Each device which has EEPROM must also have support hardware for remote programming via the serial BUS. Use of this opcode halts normal BUS activity and use of the BUS is restricted to MEMACS, MEMADD, MEMFRA, and DEVJSR (if other opcodes appear on the bus, the response may be unpredictable). Note that GROUP = 000 allowing all devices on the bus to hear it. All subsequent MEMADD and DEVJSR opcodes are directed to the device specified in the third byte of the MEMACS message. When done with a particular device, the bus can be released for normal use, or another device specified for memory access. In a multi-group system, neither GROUP nor DEVICE ADDRESS in the third byte will be sent as 0. A feature has been included to allow the data and address subsequently sent on the bus to be encrypted.

It is **highly recommended** that for programming whole blocks of memory, like in the factory , that the SBEP opcode WRITE_DATA_REQ should be used for the sake of speed. If using this opcode one must pay closeattention to the D1 and the D8 bits. Programming the memory, while the target is in normal operating mode, can cause the radio to malfunction, because itmight try to access the codeplug while its being programmed. In some radiosdue to the low microprocessorclock speed used, and high serial bus activity like when programming the eeprom could cause heavy burden on the resources andeventual breakdown. For this very reason there is a need to exit normal bus activity (with use of bit D1) and resetting on exit (with the use of bit D8).

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	X	X	X	X	X
D8	D7	D6	D5	D4	D3	D2	D1
Group			Device Address				
0	Opcode						
CRC							

D1: = 0 to enter normal bus activity
= 1 to exit normal bus activity

D3-D2: = 00 when modifying EEPROM
= 01 when modifying RAM
= 10 when modifying OPTION DEVICE 1

= 11 when modifying OPTION DEVICE 2

D4: = 0 when using physical memory
= 1 when using virtual memory (see note below)

D5: = 0 for standard 1 data byte MEMADD
= 1 for optional 3 data byte MEMADD

D6: = 0 to not use EEPROM hardware lock
= 1 to engage/release hardware lock upon entering/leaving

D7: = 0 to disable data/address encryption
= 1 to enable data/address encryption

D8: = 0 to RESET the system upon exiting
= 1 to enter normal bus activity without RESET

There are basically 3 types of memory access operations a device should try to support.

1. A RSS programmer for example is going to write an entire codeplug or blocks of a codeplug. In this case, normal device operations should be suspended so the device does not try to read a block that might be getting updated (this is the non real-time mode). In this case because eeprom was accessed the exit to normal activity should be via a reset.

2. A single (or at most a couple) byte(s) of codeplug needs to be updated. This might happen when the device is being tuned and a tuning value needs tobe saved to the codeplug. In this case one would not like to suspend normal operations because one would still want to be able to tune.

3. A user needs to read/modify RAM. This might be needed for tuning or debugging purposes.

Upon receiving this opcode with D1=1, devices must check to see if the address matches theirs. If not, then they are prohibited from transmitting anything on the bus (except a response to a Request) until MEMACS is again sent with D1=0 (even if they were previously the recipient of a prior MEMACS). If an address match does occur, then D2 can be examined to aid in setting up for EEPROM programming (eg. EEPROMs cannot be accessed while being programmed, so certain interrupts which read EEPROM contents may need to be turned off - channel scan for example). If EEPROM had been accessed, all devices should set themselves up to read their Option Status bytes after the next reset (in case they may have been changed). If D7=1, then the option should prepare to receive address/data information (via MEMADD) which has been encrypted based on some previously determined algorithm. Supplementary documents will contain details of the algorithm used by each option. If D6=1, then the option should disable any hardware write-protection circuitry to allow its memory to be modified.

D4 must be 0 in platforms that do not implement virtual addressing. ‘Physical’ addressing is merely the address that the device’s microprocessor accesses. ‘Virtual’ addresses are defined per platform, but in general, assignments of certain address ranges are given to

certain memory devices independently of the actual physical address. This is especially useful when accessing EEPROM data, since the interface is independent of changes to the physical memory decoder that might be made to the platform.

D5 can be used to enable the optional MEMADD protocol that allows 3 bytes of data to be read per message instead of the normal 1 byte per message. This is only valid for reading data since the desired address is specified in the MEMADD request. To write into memory, D5 must be set to 0.

Upon receiving MEMACS with D1=0 and only RAM had been accessed (no EEPROM), then D8 is examined to see if the device (if capable) should reset the system. If EEPROM had been accessed anywhere in the system, then D8 is ignored (treated as 0) and the system will be reset upon exiting. This will usually be the case if the Option Status has been modified by a programmer.

For the MIRS/LINGO radio, OPTION DEVICE 1 is the HC16 INternal Registers and the OPTION DEVICE 2 is an external UART/ACIA.

MEMADD - MEMORY ADDRESS/DATA - opcode $07

This opcode provides for the capability to change up to 64K of memory per address directly from the serial BUS. Before this opcode can be used, the MEMACS opcode must be sent to indicate which device address is to be accessed by subsequent MEMADD opcodes. The addresses specified by MEMADD are absolute and can contain different information from radio to radio. Thus only programmers or devices with intimate knowledge of the destination device software structure should use this opcode. It should be noted that this is one of the few opcodes that doesn't use the GROUP/ADDRESS bits. See Section 3.3 regarding Request opcodes.

Bit

7	6	5	4	3	2	1	0
Data 1							
Data 2							
Data 3							
B/R	Opcode						
CRC							

DATA 1: Contains the most significant byte of the absolute address in memory which is to be accessed. Optionally this can contain the contents of the specified address +2 (see MEMACS).

DATA 2: Contains the least significant byte of the absolute address in memory which is to be accessed. Optionally this can contain the contents of the specified address +1 (see MEMACS).

DATA 3: Contains the data to be written to or read from at the specified address.

To write into memory, a device sends a Broadcast MEMADD (B/R=0) with the appropriate address and data. To read memory contents, a device sends a Request MEMADD (B/R=1) setting the address of the desired information. The destination device then sends the Broadcast MEMADD (B/R=0) with the previously specified address and its associated data.

When using the optional MEMADD, three bytes of data can be sent instead of only one. However, the originating device must remember what address was requested since the address information is not sent along with the data. For this reason, the optional MEMADD

is only valid for memory reads. To write, the destination address must still be specified in the normal MEMADD Broadcast.

MEMFRA - MEMORY ADDRESS/DATA FRAMED - opcode $03

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 3/11/91 WITH VERSION 5.0

This Framed opcode provides for the capability to change up to 64K of memory per address (in up to 256 byte blocks) directly from the serial BUS. Before this opcode can be used, the MEMACS opcode must be sent to indicate which device address is to be accessed by subsequent MEMFRA opcodes. The addresses specified by MEMFRA are absolute and can contain different information from radio to radio. Thus only programmers or devices with intimate knowledge of the destination device software structure should use this opcode. It should be noted that this is one of 2 opcodes that doesn't use the GROUP/ ADDRESS bits. See Section (none) regarding Framed opcodes and Section 3.3 regarding Request opcodes.

Bit

7	6	5	4	3	2	1	0
Data 1							
Data 2							
Data 3							
B/R	Opcode						
CRC							

DATA 1: Contains the most significant byte of the absolute location in memory where the block of data is to be written to or read from.

DATA 2: Contains the least significant byte of the absolute location in memory.

DATA 3: Contains the number of bytes to be written to or read from starting at the specified address.

The address specified is the beginning of a block of data that extends up to an address equal to the starting address + number of bytes - 1. If the number of bytes (DATA 3) =0, then it is treated as 256 bytes. Thus if DATA 1 = DATA 2 = DATA 3, then the block extends from $0000 to $00FF inclusive. To write into memory, a device sends a Broadcast MEMFRA (B/R=0) with the appropriate address and number of bytes to be sent. This is then followed by the specified number of bytes (each message has up to 4 bytes of data and a CRC). To read memory contents, a device sends a Request MEMFRA (B/R=1) setting the address of the desired information and the number of bytes to be read. The destination device then

sends the Broadcast MEMFRA (B/R=0) with the previously specified address and its associated data.

METINF - METROCOM INFORMATION - opcode $4F

This opcode is broadcast by an option or radio to signal the change in the status of some Metrocom function. Metrocom uses this opcode for updating the control unit Early/Late status display and for updating the alarms mask in the alarms board.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains sub-opcode dependent on the source address.

DATA 2: Contains data dependent on the source address.

The address field is the sending device SOURCE address. This is important because devices receiving this opcode use the address to tell them how to interpret the data. Metrocom uses this information as shown below:

SOURCE ADDRESS	DATA 1	DATA 2	Function or Description:
01	00	0/1	Autotracking register off/on
01	01	0/1	Entering/Exiting Fallback
01	02	-	Request for location update (PRTT)
04	00	-	Turn Early/Late display off
04	01	sign magnitude	Display Early/Late as +/-nn
04	03	sign magnitude	Display Early/Late as nnE/L
07	select	mask	Data 1 - enable mechanical alarms Data 2 - mask mechanical alarms

MODUPD - MODE UPDATE - opcode $5C

This opcode was created to accommodate modes greater than 255. This one opcode, replaces the five mode update opcodes, ACTMDU, ACTMDW, TRKMDU, RXMODE and TXMODE. The option bits however have been included in a separate opcode. Therefore devices using this opcode for more than 255 modes will have to broadcast two opcodes as opposed to one, for the other mode update opcodes. The maximum limit to the number of modes is now (2^13) -1 = 8192 - 1 or 8191 and the type of mode update is defined by the Sn bits.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
S3	S2	S1	M13	M12	M11	M10	M9
M8	M7	M6	M5	M4	M3	M2	M1
0	Opcode						
CRC							

S3	S2	S1	- These bits define the type of mode update.
0	0	0	ACTMDU
0	0	1	ACTMDW
0	1	0	RXMODE
0	1	1	TRKMDU
1	0	0	TXMODE

M13 ---- M1 - contains the 13 bit Mode Number

When using this opcode individual device S/W should ensure that the appropriate option information be applied to the active mode information received i.e. wait for the OPTUPD (with the option bits) to arrive before processing the opcode. Also all devices should broadcast the OPTUPD opcode with the MODUPD opcode even if the option bits are don’t cares.

OPTSTS - OPTION STATUS VALUE - opcode $16

This opcode is used by options to obtain status information from remote EEPROM. Option Status contains option dependent information such as default values, ID's, or operating features. Each device address has 8 bytes of EEPROM storage available to it organized as 4 blocks of 2 bytes each. It is accessed by Requesting a block. Option Status information should normally be saved in the option RAM. On power-up, if the RAM was determined to be invalid, then OPTSTS should be requested after RADRDY. Options can not update their Option Status value in remote EEPROM by initiating a Broadcast with this opcode. See Section 3.3 regarding Request opcodes.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
B/R	Opcode						
CRC							

DATA 1: (B/R=1): not used
(B/R=0): Contains the first byte of the Option Status block

DATA 2: (B/R=1): Contains the block number, 0 thru 3
(B/R=0): Contains the second byte of the Option Status block

To read Option Status (B/R=1), the option sends OPTSTS specifying which of the 4 blocks (2 bytes each) it wishes to read. The response to this request is an OPTSTS broadcast (B/R=0) with the data contained in DATA 1 and DATA 2. During the Broadcast (B/R=0), GROUP/ADDRESS refers to the destination device, not the source device (the normal case).

It should be noted that if this opcode is sent with GROUP=0, then all radios will respond in a multi-radio system.

OPTUPD - OPTION INFO UPDATE - opcode $5D

This opcode was created because MODUPD opcode which when modified for greater than 255 modes could not accommodate the option bits. This one opcode, will be broadcast when ever the MODUPD opcode is transmitted by any device. Therefore devices using this opcode for more than 255 modes will have to broadcast two opcodes as opposed to one, for the other mode update opcodes. The type of mode associated with the option bits is defined by the Sn bits.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
S3	S2	S1				D1	D2
Data 1							
0	Opcode						
CRC							

S3 -	S2 -	S1	- These bits define the type of mode update.
0	0	0	ACTMDU
0	0	1	ACTMDW
0	1	0	RXMODE
0	1	1	TRKMDU
1	0	0	TXMODE

DATA 1 - Contains the Option Enables which relate to the mode being updated. The Option Enables are one byte of radio EEPROM information for each mode which are used by options for enabling certain option features on a mode- by-mode basis. Options interpret these bits based on their Option Status bytes. Each option can assume that its bits are contiguous but relocatable (the Option Status register must indicate which bits to use).

When using this opcode individual device S/W should ensure that the appropriate option information be applied to the active mode information received i.e. wait for the OPTUPD (with the option bits) to arrive before processing the opcode. Also all devices should broadcast the OPTUPD opcode with the MODUPD opcode even if the option bits are don’t cares.

<u>ACTMDW Specifics:</u>

D1 - Set this bit to have the Active Mode (specified in the corresponding MODUPD message) updated only on system reset.

D2 - Set this bit to have the Active Mode (specified in the corresponding MODUPD message) updated on transmitter de-key or system reset.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Active Mode is updated to the operator selected value. If DATA 3 equals zero or is an invalid channel number, then the mode will not change, although D1 and D2 will still be acted on. Options utilizing this opcode may want to monitor ACTMDU to make sure the radio reached the desired channel before proceeding with their normal operation. This will also allow options which use ACTMDW to determine when another option has usurped their control. In this case, options should suspend or abort their operation until the mode has been restored (ACTMDW with D1=D2=0). Options may also be designed to use ACTMDW only if it is currently “available”. Currently there are no priorities associated with ACTMDW.

OSCVAL - OSCILLATOR VALUE - opcode $43

This opcode is broadcast by an option to fine-tune the TX/RX frequency which the existing mode uses. Neither the Active mode number, nor the Selected mode number changes, although the actual RF frequency does. Since different radios have different numbers of bits to represent the local oscillator frequency and each increment is a different frequency change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure. This opcode is executed immediately upon receipt, regardless of the state of RADKEY. The typical use for this opcode is to ”warp” the reference oscillator onto the desired frequency. Thus, minute fine-tuning can be achieved. Generic options wishing to change RF frequency should use CHLNUM.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the MSB of the Oscillator Value

DATA 2: Contains the LSB of the Oscillator Value

The frequency specified by this opcode remains valid until 1) Active mode changes, normally due to ACTMDW and Button opcodes, or 2) System Reset. In other words, it is treated as if the ”D1-D2 inhibit bits” are set such that D1=0 and D2=1.

PALIM - PA LIMIT VALUE - opcode $42

This opcode is broadcast by an option to change the value of the parameters which limit transmitter output power. Since different radios have different parameters, numbers of bits to represent the parameters, and each increment is a different parameter change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure. General purpose options which desire to modify the transmit power value should use ACPRVL. This opcode is executed immediately upon receipt, regardless of the state of RADKEY. It should be noted that the actual power level obtained will be the minimum of 1) the value requested by the operator or options (via ACPRVL), or 2) hardware/software limits imposed by the radio (of which PALIM plays a part).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the MSB of the Power Limit

DATA 2: Contains the LSB of the Power Limit

The limit specified by this opcode remains valid until System Reset. Typically DATA 1 may represent a voltage limit while DATA 2 may represent a current limit.

PLDECT - PL DETECT - opcode $23

This opcode is Broadcast by the radio to inform options of a PL code detection. On carrier squelch modes, this opcode will not be sent. On coded squelch modes (as defined by the value in Active RX PL Code), this opcode will be sent even if PL is not being used in the muting decision (e.g. HUB Off-Hook).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	D2	D1
0	Opcode						
CRC							

D1: = 0 when coded squelch has been lost or Reverse Burst (TOC) has been detected.
= 1 when coded squelch is present.

D2: = 0 when audio will be muted due the channel being unqualified.
= 1 when audio will be unmuted due to a qualified channel.

D2 can be considered an Audio Detect bit. It indicates when the radio wants to unmute audio due to detecting a qualified channel. Since PLDECT is only sent out on PL sensitive channels, Audio Detect (D2) is also sent with the SQLDET opcode. Audio Detect is not only sensitive to PLDECT and SQLDET, but also to the state of the Hangup Box (HUB), RPLEN, and the PL muting/unmuting types (eg. AND/OR/PCI in Syntor X). The particular role each of these has in determining Audio Detect may change from model to model and should be described in supplementary documents. It should be pointed out that even though D2=1, audio may still be muted due to options such as Priority Scan mute, Selective Call muting, or the presence of an invalid Securenet code.

It is envisioned that Audio Detect be used by options that need to know about the presence of a valid channel without requiring them to keep track of the coded squelch operation of the radio (which can change from mode to mode). A vehicular repeater is such an option.

PRTVAL - PORT VALUE - opcode $44

This opcode and its associated values are used to monitor the state of a device. Parallel I/O ports, internal RAM, or other byte-wide locations can be read via this opcode without halting normal bus activity or radio operation. Up to 64 locations (hereafter called “ports”) per device can be monitored. Provisions have been also made to initiate the condition where port values can be periodically transmitted on the bus automatically. Since different devices have different bit assignments on its ports, general purpose options should refrain from using this opcode. It is envisioned that this opcode will be used mainly for diagnostics or by options intimate with the device software/hardware.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
D2	D3	Data 4					
0	Opcode						
CRC							

DATA 1: Contains the port value (0 thru 255)

D2: = 0 to cause the device to read the port data
= 1 when the device responds with the port data

D3: = 0 to cause the port to be read once
= 1 to cause the port to be read periodically until halted

DATA 4 - Contains the port number ($00 thru $3F)

To cause a device to read a port, send D2=0 with the appropriate D3 and DATA 4. Both GROUP and DEVICE ADDRESS must specify the destination address while DATA 1 will be ignored. When the port data is ready (eg. A/D conversion is complete), the device will respond with D2=1, DATA 1 = port value, and DATA 4 = port number. Both GROUP and DEVICE ADDRESS will then specify the source address as normal while D3 may contain unspecified data. Note that a request version of this opcode has not been defined. This is because the time it takes to formulate the specific port value may exceed the time allowed for a response to the request. If D3=1 had been sent to cause periodic port updates, the updates will stop only when this opcode is again received with D3=0 or upon system reset. Timing of the periodic updates will be a function of the particular implementation and may

vary from port to port. If a device does not implement a particular port, then any attempts to access it should be ignored.

PRUPST - POWER UP STATUS - opcode $3B

This opcode is normally Broadcast by each device on power up indicating the result of local self-check routines. It can also be sent any other time a device detects an error condition. If any device in the system has a fatal error, the display will show an appropriate error message and the radio will become unusable. See Section 3.4.1 for a discussion on power up considerations. Note that GROUP = 000 so that all devices will hear it.

Bit

7	6	5	4	3	2	1	0
0	0	0	X	X	X	X	X
Data 1							
Group			Device Address				
0	Opcode						
CRC							

BIT	SIGNIFICANCE	SUGGESTED TEST
0	ROM FAIL	CHECKSUM
1	EEPROM FAIL	CHECKSUM
2	EEPROM BLANK	$FF CHECK
3	RAM FAIL	INVERT
4	HARDWARE FAIL	any
5	SERIAL BUS ERROR	re-transmitted 8 times
6	KEYPAD LOCKED	no test
7	FATAL ERROR	no test

Each bit of DATA 1 is set to 1 when an error condition occurs as described above. An example of a hardware failure is a DVP encryption key failure (a fatal error - the operator cannot use the radio without secure communications). The keypad locked bit can be used for security applications. It informs the radio that the control head is locked and to keep audio muted and (perhaps) not transmit until PRUPST is sent with GROUP/ADDRESS = $05 and keypad unlocked. The keypad can be unlocked with a proper sequence of key presses (these are not sent on the bus, only the control head reads these).

Supplemental documents will describe the details of hardware checks and fatal error conditions that a particular device has as well as how these bits are implemented.

Upon receipt of an error which causes the system to reset, the actual reset should be delayed long enough for the display to show the error message.

PTTINH - PTT INHIBIT - opcode $18

This opcode is Broadcast by options to inhibit radio key up due to PTT inputs. It also can be used to keep other options from using PTT. On system RESET, PTT is not considered inhibited although the radio may still not key on a PTT press due to parameters in the radio EEPROM.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	D2	D1
0	Opcode						
CRC							

D2	D1	
0	0	PTT is enabled
0	1	The Radio is PTT disabled
1	0	Not Allowed, this is an invalid state
1	1	All devices are PTT inhibited (except the device broadcasting this opcode)

Options may set PTT Inhibit for several different reasons. For example, the Siren Public Address option sets it because PTT presses from the operator indicate he wishes to talk through the Siren PA instead of keying the radio. However, an Intercom unit for EMS (Emergency Medical Service) may set it so that it could use the microphone to allow communication between the front and rear cab of a vehicle. If these two options were in the same system some considerations must be made. If Siren PA sends out a PTTINH, it is desired that the Intercom not send the microphone audio to the rear cab. Thus, Siren PA needs to send out PTTINH (D2=1) whenever it needs it, even if the Intercom had already sent it. This allows the Intercom to know that another option is disabling PTT and that it, too, must ignore PTT. Therefore, options should follow these two rules: 1) An option which reacts when PTT is pressed should keep track of the D1 bit so that it does not react when another option has PTT transmissions inhibited (ie. when D1=1), and 2) When an option wants PTT inhibited (D2=1), it should broadcast a PTTINH even if D2 is already set. Note that, in the above example, there will be a slight glitch when the PA is turned off. The Siren will uninhibit PTT allowing the radio to key. The Intercom is still on and so it will again inhibit PTT. During the length of time it takes to do this, the radio could be starting its key-

up sequence if PTT was pressed. This also highlights a third rule: 3) If PTT becomes uninhibited (D2:D1 = 01 or 00), options should check to see if they are in a mode that allows this, otherwise, they should again send PTTINH (with D2:D1 = 11 or 01).

GROUP address is always sent as 000 to guarantee that all options see it.

PWRCTL - POWER CONTROL - opcode $48

WARNING - THIS OPCODE HAS BEEN OBSOLETED STARTING ON 12/9/92 WITH VERSION 5.1

This opcode is Broadcast whenever it is desired to put one or more devices into a ”reserve power” or ”power down” state. Control head display dimming can also be controlled. There are eight power down states and four display brightness levels that can be specified. Only one device per installation may Broadcast this opcode.

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Group			Device Address				
X	X	X	X	X	X	Data 2	
Data 1							
0	Opcode						
CRC							

DATA 1: Contains the power control level. Each bit location corresponds to an increasing savings in power according to the table below:

bit 0: slight savings (eg. display off)
.
.
bit 7: greatest savings (eg. complete power off)

DATA 2: 00 for bright display
01 for medium display
10 for low display
11 for off display

The GROUP and DEVICE ADDRESS may specify a particular device or the entire system (GROUP = DEVICE ADDRESS = 0). If an option has no control over its power condition, it may ignore this opcode. For backward compatibility, whenever a bit is set in DATA 1, then all lower-order bits should also be set. The action taken by an option for each power level can be found in the supplement describing that option. It should be noted that, in a sense, this opcode uses negative logic. For example, when DATA1 bit0=1, then a slight savings is to be activated and the display turned OFF.

RADKEY - RADIO KEYED - opcode $19

This opcode is Broadcast by the radio to inform other devices that the radio has been keyed and that RF power is now present or that the radio has been de-keyed. Broadcasts generated external to the radio (eg. multi-radio systems) will not effect the radio. Devices which utilize the transmitter should wait for this opcode and not rely on TXCTRL or PTT presses because PTTINH and variable transmitter rise times make it hard to predict when to start transmission. This opcode is also sent when the TOT has expired. This can be treated as a warning for the options to allow them time to gracefully close their operation (eg. DVP may send EOM, MDC may send ID). The warning will typically treated as a PTT release.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
D1	X	X	X	X	X	X	D2
0	Opcode						
CRC							

D1: = 0 when the radio is keying up or down
= 1 when the radio is about to key down due to TOT

D2: = 0 when radio not keyed
= 1 when radio is keyed

This opcode and its data is sent in the following situations:

D1	D2	Situation
0	0	Radio is keying down
0	1	Radio is keying up
1	0	Never happens (radio about to key-up?)
1	1	Radio is about to key down

RADRDY - RADIO READY - opcode $15

This opcode is sent only after power up by the radio to indicate that normal BUS activity is allowed for that GROUP (See Section 3.4.1). Until RADRDY is sent, the radio will remain muted and de-keyed. Only PRUPST or any response to a Request are allowed on the BUS (exception: the control head can also send SETBUT and BUTCTL). After RADRDY is sent, normal bus activity commences for that GROUP. RADRDY will be transmitted only when PRUPST for each address in the GROUP has been received (ie. the radio must know what addresses are occupied in its GROUP) or when the power-up timer has expired.

Since some bus options are removable and can be re-connected without first powering-down the radio, it is necessary for the radio to send a RADRDY message each time it receives an unexpected PRUPST from an option. ‘Unexpected’ refers to those that are received after the normal power-up period when each option reports it’s PRUPST prior to the first RADRDY that the radio sends.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							p/m
0	Opcode						
CRC							

DATA 1: Contains data pertinent only to that radio model.

DATA 2: d7 - d1 contains data pertinent only to that radio model.
d0: Portable / Mobile bit (Portable = 0, Mobile = 1)

Note that DEVICE ADDRESS will always be 1.

RCLBUT - RECALL BUTTON - opcode $10

This opcode is broadcast by control heads to indicate that the user wishes to review a list, such as an operator selected scan list. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

REQBUT - REQUEST BUTTON - opcode $11

Control heads can Broadcast this opcode instead of EXTBUT to indicate that the user wants to exit the configuration mode of operation because the option may not yet be ready. Once the option becomes ready (eg. a data option has received an ACK), it may enable the control head to exit configuration by Broadcasting EXTBUT up to the control head using its source GROUP and ADDRESS. Configuration mode is not exited until the control head responds with its own EXTBUT. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

DATA 2: Contains any data associated with the button

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

REVINH - REVERSE BURST INHIBIT - opcode $2C

This opcode is Broadcast by options to prevent the transmission of Reverse Burst or Turn Off Codes (TOC) upon transmitter de-key of PL encoded transmissions.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	x	x	x	D3
0	Opcode						
CRC							

D1: Set this bit to have Reverse Burst Inhibit (D3) cleared on system reset.

D2: Set this bit to have Reverse Burst Inhibit (D3) cleared on transmitter de-key, Active Mode changes, or system reset.

D3: = 0 to not inhibit Reverse Burst (or Turn Off Code)
= 1 to inhibit Reverse Burst (or Turn Off Code)

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, Reverse Burst (TOC) enable is updated to the radio EEPROM state.

RPTDIR - REPEAT/DIRECT - opcode $24

This opcode is broadcast by options to select between two transmit frequencies stored in radio EEPROM. It should not inhibit an operator from selecting another state nor should the opcode affect the operator selected state (ie. the radio should remember both the option defined state as well as the operator selected state). This opcode should only be used in an attempt to change the state of RPT/DIR. Use ACTSTU to monitor the current state of RPT/DIR. The main frequency will nominally be the repeated frequency and the alternate frequency will nominally be the direct frequency.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	x	x	x	D3
0	Opcode						
CRC							

D1: Set to have D3 (RPT/DIR) updated only on system reset.

D3: Set to have D3 (RPT/DIR) updated on Active Mode Changes

D3: = 0 to select the main frequency (RPT).
= 1 to select the alternate frequency (DIR).

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Repeat/Direct state is updated to the radio EEPROM or operator selected state. If an alternate transmit frequency is not enabled on a mode and D3 = 1, the radio will not transmit. Although this opcode is normally used to switch between Repeater and Direct transmit frequencies, the two transmit frequencies can be used for other purposes.

RXAUD - RECEIVER AUDIO ROUTING - opcode $1A

This opcode allows options to manipulate the receive audio path from the discriminator to the speaker. It is implemented by the radio. Audio will be muted during alert tones regardless of the state of RXAUD.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	x	x
D1	D2	Data 3			Data 4		
0	Opcode						
CRC							

D1: = 0 to let radio determine audio muting
= 1 to force the audio to unmute

D2: = 0 to not priority sample receive audio routing
= 1 to enable priority samples to mute receive audio routing

Data 3: = PRIORITY = 000 to turn off
= 001 for lowest priority
= 010 for next lowest priority
.
= 111 for highest priority

Data 4: = ROUTING = 000 to disconnect the RXAUD line
= 001 to connect to audio shaping circuitry
= 010 to connect to attenuator input
= remainder reserved

RXAUD ROUTING can always be used unless an option of higher priority is using it. Thus options should always monitor this parameter and turn their audio off when another option uses it. To release the audio lines and make them available, set D1, D2, D3, and D4 to 0. Typical priorities are: tones > MVS > DVP. The radio does not check for violations of priority. Whenever D4 =/ 000 the discriminator is automatically muted. When D4 = 000, discriminator muting is controlled by D1 and D2. Thus, to force discriminator output, set D4=000 and D1=1. D1=0 is useful for options that want to control the audio signal, but want the radio to control the muting, such as a voice scrambler or DVP. These same options will

usually set D2=1 to keep priority sample noise bursts from being deciphered. See Section 2.2 for a description of the audio lines. D1, D2, D3, and D4 are set to 0 on system reset.

When choosing a priority for an option, supplements of the existing options should be utilized to ensure compatibility.

RXMODE - RECEIVED MODE UPDATE - opcode $21

This opcode is Broadcast by the radio when the received mode has changed. On Active Mode changes, ACTMDU will be Broadcast but RXMODE will not be Broadcast until it differs from the Active Mode (eg. scan is turned on). It also contains the Option Enables for the received mode.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the Option Enables which relate to the Received Mode. The Option Enables are one byte of radio EEPROM information for each mode which are used by options for enabling certain option features on a mode- by-mode basis. Options interpret these bits based on their Option Status bytes. Each option can assume that its bits are contiguous but relocatable (the Option Status register must indicate which bits to use).

DATA 2: Contains the Received Mode number.

During channel scan, if no channel is qualified, then RXMODE will be broadcast with DATA 1 = 0 and DATA 2 = 0.

RXPLIN - RECEIVE PL MUTE CONTROL INHIBIT - opcode $2B

This opcode allows an option to put the radio in a receive carrier squelch state without disabling any data decoder (such as PL). It also affects any internal scan dependent feature by causing the scan to lock onto the existing channel regardless of whether any qualifying data is present or not (squelch still has to be active, however). Using this opcode is equivalent to taking the HUB (Hang-Up Box) On or Off-Hook. The radio will also broadcast this opcode to reflect the state of its internal data decoder after being changed by a monitor button on the control head or by the HUB. This can then be used by other options (eg. Selective Call) in determining what state they should be operating in (eg. if the monitor button is on, then Selective Call may want to be disabled as well as the PL decoder in the radio).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	D1
0	Opcode						
CRC							

D1: = 0 to allow the PL decoder to operate normally
= 1 to inhibit the PL decoder from affecting audio muting

Even when D1 is set, PL Detect is still Broadcast. However, the result is not used in MUTE control. This bit is cleared by the radio on system reset.

SCONOF - SCAN ON/OFF (ALL SCANS) - opcode $2D

This opcode is Broadcast by options to change the state of the Channel Scan On/Off control. It should not inhibit an operator from selecting another state nor should the opcode affect the operator selected state (ie. the radio should remember both the option defined state as well as the operator selected state). This opcode should only be used in an attempt to change the state of Scan On/Off. Use ACTSTU to monitor the current state of Scan On/Off.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	x	x	x	D3
0	Opcode						
CRC							

D1: Set this bit to have D3 (Scan On/Off) updated only on system reset.

D2: Set this bit to have D3 (Scan On/Off) updated on transmitter de-key, Active Mode changes, or system reset.

D3: = 0 to turn off Channel Scan
= 1 to turn on Channel Scan

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, Scan On/Off is updated to the radio EEPROM or operator selected state.

SETBUT - SET BUTTON - opcode $0A

This opcode is Broadcast by control heads to indicate that the user has selected a value such as which mode to operate on, for example. It is also used to indicate the state of the various switches in the system. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

DATA 2: Contains the data associated with the button

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case) except when referring to the system BUTTON REGISTERS ($00 thru $0F). In this case, GROUP/ADDRESS = $00.

If possible, all options should be configurable using SETBUTs, as well as ENTBUTs and EXTBUTs. This will allow them to be useable with more conventional push-button style control heads (as opposed to menu- driven control heads).

SFTPT1 - SOFT POT 1 - opcode $56

This opcode is broadcast to adjust various programmable hardware parameters. Neither the Active mode number, nor the Selected mode number changes. General purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio hardware structure. This opcode is executed immediately upon receipt, regardless of the state of RADKEY. The typical use for this opcode is to tune or adjust the hardware to meet specific requirements. If the soft pot register is requested, the returned data is the value currently in use which may be different from the corresponding EEPROM value.

Bit

7	6	5	4	3	2	1	0
Group			D1	X	X	X	X
Data 1							
Data 2							
B/R	Opcode						
CRC							

D1: =0 program soft pot however do not save soft pot data to EEPROM.
=1 program soft pot and save soft pot data to EEPROM.

DATA 1: Register of Soft Pot: (Refer to the appropriate product specific documentation for definitions).

DATA 2: Contains the right justified data to be loaded into the soft pot.

SGINFO - SIGNALLING INFORMATION - opcode $25

This opcode is Broadcast by a signalling option to indicate that a valid signal has been detected. It is Broadcast on the bus to allow other, more sophisticated data options to interpret the data and/or command even if the original data option cannot. Each byte of the data signal, as well as its position in the data stream, is put on the bus for interpretation by other options. This opcode also allows external options to command a signalling option to transmit its own data.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
B/R	Opcode						
CRC							

DATA 1: Contains the sequence number of the byte in the data stream (1=1st byte, 2=2nd byte, etc).

DATA 2: Contains the data.

Upon receipt of a complete message, a signalling option may Broadcast (B/R=0) this opcode to let other options know what it has received. The first byte is broadcast with DATA1=1, the second byte with DATA1=2, etc. The end of the message is determined by the contents of the signalling data itself. Only actual data is sent in this opcode. Any coding (CRC, interleaving, etc) overhead is stripped off before the signalling data is Broadcast. At times it may be desirable for the signalling option to Broadcast this opcode only when the signalling word ID matches its own ID. At other times, this opcode should be sent regardless of the ID specified by the signalling data. Each case should be specified in the appropriate supplement for each signalling option.

A non-signalling option may send some this opcode with B/R=1 to cause the signalling option to transmit any data it wishes. The actual RF data packet won’t be sent until all required bytes have been transferred from the non-signalling option to the signalling option via this opcode. Note that since an end-of- message indicator is not provided by this opcode, the signalling option must determine when all bytes have been transferred by

interpreting the signalling data itself. Only actual data is sent in this opcode. Any coding overhead (CRC, interleaving, etc) is provided by the signalling option.

SHOBUT - SHOW BUTTON - opcode $09

This opcode is broadcast by control heads to indicate that the user wishes to see a value such as volume, for example. See Section 3.5.1 for a discussion on button handling.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

DATA 1: Contains the button register associated with that particular button and that particular device.

Both the GROUP and DEVICE ADDRESS specify the destination address instead of the source address (the normal case).

SQLDET - SQUELCH DETECT - opcode $1E

This opcode is Broadcast by the radio to indicate a change in the carrier detect state and is independent of the state of the HUB and Monitor button. Externally generated Broadcasts (eg. multi-radio systems) have no effect on the radio.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	D2	D1
0	Opcode						
CRC							

D1: = 0 when carrier is not present
= 1 when the long time constant carrier is present

D2: = 0 when audio will be muted due the channel being unqualified.
= 1 when audio will be unmuted due to a qualified channel.

D2 can be considered an Audio Detect bit. It indicates when the radio wants to unmute audio due to detecting a qualified channel. Since audio muting may also be sensitive to PL channels, Audio Detect (D2) is also sent with the PLDECT opcode. Audio Detect is not only sensitive to PLDECT and SQLDET, but also to the state of the Hangup Box (HUB), RPLEN, and the PL muting/unmuting types (eg. AND/OR/PCI in Syntor X). The particular role each of these has in determining Audio Detect may change from model to model and should be described in supplementary documents. It should be pointed out that even though D2=1, audio may still be muted due to options such as Priority Scan mute, Selective Call muting, or the presence of an invalid Securenet code.

It is envisioned that Audio Detect be used by options that need to know about the presence of a valid channel without requiring them to keep track of the coded squelch operation of the radio (which can change from mode to mode). A vehicular repeater is such an option.

SQLVAL - SQUELCH VALUE - opcode $28

This opcode is Broadcast by options to modify the squelch setting of the radio. It should not inhibit an operator from selecting another value nor should the opcode affect the operator selected value (ie. the radio should remember both the option defined value as well as the operator selected value). Since different radios have different numbers of bits to represent squelch and each increment is a different squelch change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	Data 3			
0	Opcode						
CRC							

D1: Set this bit to have DATA 3 (Squelch Value) returned on system reset.

D2: Set this bit to have DATA 3 (Squelch Value) returned on Active Mode changes or system reset.

DATA 3: Contains the squelch value.
= 0 for unsquelched
= $F is for tight squelch.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Squelch Value is updated to the radio EEPROM value or the operator selected value.

SQLVAL - SQUELCH VALUE - opcode $28

This opcode is Broadcast by options to modify the squelch setting of the radio. It should not inhibit an operator from selecting another value nor should the opcode affect the operator selected value (ie. the radio should remember both the option defined value as well as the operator selected value). Since different radios have different numbers of bits to represent squelch and each increment is a different squelch change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	D1	D2
X	X	X	X	Data 3			
0	Opcode						
CRC							

D1: Set this bit to have DATA 3 (Squelch Value) returned on system reset.

D2: Set this bit to have DATA 3 (Squelch Value) returned on Active Mode changes or system reset.

DATA 3: Contains the squelch value.
= 0 for unsquelched
= $F is for tight squelch.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Squelch Value is updated to the radio EEPROM value or the operator selected value.

SYSTAT - SYSTEM STATUS - opcode $4D

This opcode is Broadcast by the trunking option to inform options of the last system status OSW received over the trunking control channel. It is sent when the home system's system status OSW changes when system scan is on. This is one of the few opcodes that doesn't use the ADDRESS bits.

Bit

7	6	5	4	3	2	1	0
Group			0	0	0	0	D1
Data 1							
Data 2							
0	Opcode						
CRC							

D1: Contains system status OSW I bit.

DATA 1: Contains the MSB of the system status OSW

DATA 2: Contains the LSB of the system status OSW

The 16 bit value has the same format that is in the Smartnet Type II Protocol Specification document. A data option, for example, will use the Data Mode Fully/Reduced Operation bit. In conventional or failsoft operation the trunking option will send the System Status OSW opcode with all 17 bits cleared (this is an invalid system state since the three MSBs cannot be clear due to system ID confusion).

TBONOF - TALKBACK ON/OFF - opcode $31

This opcode allows options to change the state of the Talkback On/Off bit. It should not inhibit an operator from selecting another state nor should the opcode affect the operator selected state (ie. the radio should remember both the option defined state as well as the operator selected state). This opcode should only be used in an attempt to change the state of Talkback. Use ACTSTU to monitor the current state of Talkback.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	x	x	x	D3
0	Opcode						
CRC							

D1: Set this bit to have D3 (Talkback On/Off) updated only on system reset.

D2: Set this bit to have D3 (Talkback On/Off) updated on Active Mode changes or system reset.

D3: If D3 = 0 the Talkback feature of Channel Scan is off.
If D3 = 1 Talkback is on.

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the On/Off value is updated to the radio EEPROM value or the operator selected value.

TIMEUP - TIME UPDATE - opcode $4E

This opcode is Broadcast ONLY by the radio to send the current time-of-day or day/month/year to all serial bus options in the system. The radio receives time or date update data and must be the only source of this opcode. Despite being the sole source of time updates, the radio will most likely not be the system time keeper. Time-of-day, will be counted and calculated in every option that requires it. Examples are the Metrocom control unit and location processors. TIMEUP are typically sent once every hour.

Bit

7	6	5	4	3	2	1	0
Group			0	0	D1	D2	D3
HOUR				MINUTES MSB			
MIN LSB		SECONDS					
0	Opcode						
CRC							

D1: = 0 to indicate AM or PM on Time display (1 to NOT indicate AM or PM)

D2: = 0 to show time in 12 hour format (1 to show time in 24 hour format)

D3, HOUR:= Binary representing of the hour (1-24). D3 is the most significant bit.

MINUTES:= Binary representation of minutes (0-60).

SECONDS:= Binary representation of seconds (0-60).

The fact that there is no address field should not cause any system problems since the radio is the only source of this opcode.

TOTVAL - TIME-OUT-TIMER VALUE - opcode $36

This opcode allows options to change the radio transmitter Time Out Timer value. It should not inhibit an operator from selecting another state nor should the opcode affect the operator selected value (ie. the radio should remember both the option defined value as well as the operator selected value). A new TOT value defined by this opcode is used immediately. If the current timer value exceeds the value specified by this opcode, then time out will occur immediately.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	Data 3						
0	Opcode						
CRC							

D1: Set this bit to have the Time Out Timer value updated only on system reset.

D2: Set this bit to have the Time Out Timer value updated on transmitter de-key, Active Mode changes, or system reset.

DATA 3: Contains the number of 5 second (+/- 10%) intervals which make up the time-out time.

If DATA 3 = 0 the time out time is infinite (effectively disabled).

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the TOT Value is updated to the radio EEPROM value or the operator selected value.

TRKMDU - TRUNKED MODE UPDATE - opcode $46

This opcode is broadcast by the radio whenever the Active mode changes to a trunked mode due to operator Button Press or an ACTMDW opcode. It also contains the Option Enables for the Active Mode. Options cannot change modes by Broadcasting this opcode, they must use ACTMDW. If the Active mode is a conventional non- trunked mode, then ACTMDU will be sent instead of TRKMDU.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the Option Enables which relate to the Active Mode. The Option Enables are one byte of radio EEPROM information for each mode which are used by options for enabling certain option features on a mode- by-mode basis. Options interpret these bits based on their Option Status bytes. Each option can assume that its bits are contiguous but relocatable (the Option Status register must indicate which bits to use).

DATA 2: Contains the Active Mode Number

If an option is valid only for conventional channels, it may turn itself off while operating in a trunked system. The Option Enables may then be used by another (trunked) option.

TRKOPC - TRUNKING OPCODES - opcode $45

This opcode is Broadcast by an option in order to implement the special functions that a trunking option requires. Together with CHLNUM, an otherwise conventional radio can become a trunking radio. Even though all of the special trunking controls are available via this opcode, a hard-wire line is still necessary between the radio and the trunking option to allow the option to quickly key up/down the radio when transmitting an ISW.

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains data associated with the trunking opcode

DATA 2: Contains the special trunking opcode

See APPENDIX G for a description of the Syntor X Trunking option sub-opcodes.

TSTMOD - TEST MODE - opcode $40

Note: There is a very similar opcode named 'TESTOP' ($5B). The main difference is that 'TSTMOD' does not have a 'device address' field (and hence implies the radio address $01). The 'TESTOP' opcode has a 'device address' field and can be used by any bus option or radio. Prior to 14 Jan '91, TSTMOD had a 'device address' field, however, the test mode data requirements of radios began to exceed 2 bytes per test mode, and the field was re-defined. Since no options have ever used this opcode, it was alright to re-define it's usage solely for radios.

This opcode is Broadcast by an option (typically a system tester/analyzer) to cause a device to execute a specific predetermined test sequence. It has no effect on any other device in the system nor does it interfere with normal bus operation. This opcode is radio specific and any device using it must be intimately familiar with the specified device.

Bit

7	6	5	4	3	2	1	0
Group			Data 3				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the MSB of the test mode to be executed

DATA 2: Contains the LSB of the test mode to be executed

DATA 3: Contains additional test mode data or sub-opcode if needed.

These tests may be transitory or continuous in nature. In the latter case, the device may return to normal operation either 1) after a RESET, 2) by receiving this opcode with DATA1=DATA2=0, or 3) some other external trigger. Descriptions of the test modes available in a particular device can be found in supplements describing that device.

TXAUD - TRANSMIT AUDIO ROUTING - opcode $1B

This opcode allows options to manipulate the transmit audio path. It is implemented by the radio.

Bit

7	6	5	4	3	2	1	0
Group	Group	Group	Device Address	Device Address	Device Address	Device Address	Device Address
D1	x	x	x	x	x	x	x
D2	D3	Data 4	Data 4	Data 4	Data 5	Data 5	Data 5
0	Opcode	Opcode	Opcode	Opcode	Opcode	Opcode	Opcode
CRC	CRC	CRC	CRC	CRC	CRC	CRC	CRC

D1: = 0 to unmute the MIC (1 to mute the MIC)

D2: = SUBAUDIO= 1 when a subaudio band (f < 300Hz) option is configuring

D3: = AUDIO: = 1 when an audio band (f > 300Hz) option is configuring

Data 4: = PRIORITY = 000 for off
= 001 for lowest priority
= 010 for next lower priority
.
= 111 highest priority

Data 5: = ROUTING = 000 to disconnect the TXAUD line
= 001 to connect to splatter filter
= 010 to connect to buffer input
= remainder reserved

When using the TXAUD line, options assign their priority and frequency band to the line. During RADKEY = 0, any option can reconfigure the TXAUD lines unless an option of the same band and higher priority has it used. During RADKEY = 1, an option can reconfigure the lines only if it has a priority higher than the option in the same band which is currently using it. Options must always monitor the state of TXAUD and turn their own audio off when another option of the same band uses it. When done, options disconnect the audio line by setting D4 to 0 and D2 and/or D3 to 1 (depending on whether they are releasing subaudio and/or audio lines) ,while leaving D1 and D5 unchanged. Options of different bands may always use the lines simultaneously. If a subaudio band option writes TXAUD while an audio

band option is using it, it must not change the current value of D1 and D5. An audio band option may always redefine D1 and D5 regardless of what a subaudio band option had previously setup (except it cannot set D5=0). Whenever D2 = 1, transmit PL and reverse burst are automatically inhibited. Whenever D2 = 0, transmit PL is controlled by ACTXPL and reverse burst by REVINH. After transmitter de-key D1, D2, D3, D4, and D5 all are reset to zero.

TXCTRL - TRANSMIT CONTROL - opcode $2E

This opcode allows any number of options to key the radio without interfering with each other. The radio keeps an internal transmit counter that gets incremented and decremented as different options instruct the radio to key-up or key-down. To key the radio, TXCTRL is sent instructing the radio to increment its transmit number. As more options desire key-up, this internal number gets larger. The number is used to keep the radio keyed up until all the options which desire to transmit are done. As options get done with their portion of the transmission, they again Broadcast TXCTRL, but instructing the radio to decrement its transmit number. Only when the transmit number goes to zero (all options are done), or when TOT expires, will the radio de-key. The transmit number gets automatically cleared whenever RADKEY is sent with D2=0, after an Active Mode change, or during RESET.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	D1
0	Opcode						
CRC							

D1: = 1 to increment transmit counter
= 0 to decrement transmit counter

TXLTIN - TRANSMIT LIGHT INHIBIT - opcode $34

This opcode allows the options to inhibit the radio from lighting the transmit indicator on the control head whenever the radio is keyed either by an option or the operator.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
x	x	x	x	x	x	x	D3
0	Opcode						
CRC							

D1: Set this bit to have D3 cleared on system reset.

D2: Set this bit to have D3 cleared on transmitter de-key, Active Mode changes, or system reset.

D3: = 0 to not inhibit the transmit indicator
= 1 to inhibit the transmit indicator

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the radio reverts to not inhibiting the transmit indicator.

TXMODE - TRANSMIT MODE UPDATE - opcode $20

This opcode is Broadcast by thy radio to inform options that the mode the radio will transmit on has changed. On Active Mode changes, ACTMDU will be Broadcast but TXMODE will not be Broadcast until it differs from the Active Mode (eg. during the talkback period of talkback scan). It also contains the Option Enables for the transmit mode. Note that if the transmit mode is invalid, TXMODE may still be sent indicating the transmit mode has not changed).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DATA 1: Contains the Option Enables which relate to the Transmit Mode. The Option Enables are one byte of radio EEPROM information for each mode which are used by options for enabling certain option features on a mode- by-mode basis. Options interpret these bits based on their Option Status bytes. Each option can assume that its bits are contiguous but relocatable (the Option Status register must indicate which bits to use).

DATA 2: Contains the Transmit Mode number.

UNQSCN - UNQUALIFY SCAN - opcode $30

This opcode is Broadcast by options to force the scan to continue scanning if it has locked onto an undesirable channel. Due to the manner in which priority scans work, it is only effective with a non-priority channel. It will be used mainly by any option dependent scan.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
0	Opcode						
CRC							

VOLMIN - VOLUME MINIMUM - opcode $26

This opcode is Broadcast by options to force the volume to at least the default value. If the actual volume is larger than the default value, then no change takes place. If the actual value is smaller, then it will be increased to the default state. It should not inhibit an operator from changing his volume level nor should the opcode affect the operator selected level (ie. the radio should remember the operator selected value even when this opcode is in effect).

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	D1
0	Opcode						
CRC							

D1: = 0 to restore the volume to the previous operator selected value.
= 1 to use the system default value (stored in radio EEPROM) as the active volume if the operator selected volume is less than the the default value.

D1 is cleared on system reset. Notice that even though this opcode is changing the actual volume level, no ’D bits’ are specified. This is because is not desirable to lock the operator out since he may want to increase the volume even further.

VOLVAL - VOLUME VALUE - opcode $27

This opcode allows an option or programmer to read or write the current volume into a specified device (eg. radio). It should not inhibit an operator from selecting another volume level nor should the opcode affect the operator selected level (ie. the radio should remember both the option defined volume level as well as the operator selected level). Since different radios have different numbers of bits to represent volume and each increment is a different volume change, general purpose options should refrain from using this opcode since the effect will change across radio models. It is envisioned that this opcode will be used mainly by programmers or devices intimate with radio software structure. A limited, but radio independent, method of changing volume is to use the VOLMIN opcode.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
x	x	x	x	x	x	D1	D2
Data 3							
B/R	Opcode						
CRC							

D1: Set this bit to have DATA3 (Volume Value) updated only on system reset.

D2: Set this bit to have DATA3 (Volume Value) updated on Active Mode changes or system reset.

DATA 3: Volume value - $00 minimum value $FF maximum value

Either D1 or D2 must be set whenever an option Broadcasts this opcode. When both D1 and D2 are cleared, the Volume Value is updated to the radio EEPROM value or the operator selected value. To write a new volume value, a device sends a Broadcast VOLVAL (B/R=0) with the appropriate data. To read the volume, a device sends a Request VOLVAL (B/R=1) with the associated data. See Section 3.3 regarding Request opcodes. VOLMIN has no effect on the value of the volume contained in this opcode (ie., when reading DATA 3, the same value is obtained regardless if VOLMIN is active or not).

VRSSTU - VEHICULAR REPEATER STATUS UPDATE - opcode $4A

This opcode is sent by a vehicular repeater option to inform other devices of its various operating modes. Thus, this opcode indicates the current operating status and does not allow control of the vehicular repeater option. This allows other options to modify their operating characteristics depending on whether a vehicular repeater unit is currently on and/or transmitting.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
X	X	X	X	X	X	X	X
X	X	X	X	X	X	D2	D1
0	Opcode						
CRC							

D1: = 0 when Vehicular Repeater is Off
= 1 when Vehicular Repeater in ON

D2: = 0 when Vehicular Repeater is receiving
= 1 when Vehicular Repeater is transmitting

XFRBDY - TRANSFER DATA BODY OF SEQUENCE - opcode $51

XFREND - TRANSFER DATA END OF SEQUENCE - opcode $52

XFRERR - TRANSFER DATA SEQUENCE ERROR - opcode $53

XFRSTR - TRANSFER DATA START OF SEQUENCE - opcode $50

These opcodes are used to transfer potentially long variable length data messages to and from the radio processor (only). When an option broadcasts one of these opcodes with its DEVICE ADDRESS, the radio is automatically assumed to be the destination, and when the radio broadcasts one of these opcodes, the DEVICE ADDRESS field holds the destination option address.

Bit

7	6	5	4	3	2	1	0
Group			Device Address				
Data 1							
Data 2							
0	Opcode						
CRC							

DEVICE ADDRESS:= source address when sent by option.
= destination address when sent by radio.

DATA 1 and DATA 2 are defined below:

Opcode:	Function:	DATA 1	DATA 2
XFRSTR (50)	Start of Sequence	Offset into the Destination Buffer	# of Bytes to Follow minus one
XFRBDY (51)	Body of Sequence	Raw Data Byte	Raw Data Byte
XFREND (52)	End of Sequence	Offset into the Destination Buffer	# of Bytes Just sent minus one
XFRERR (53)	Sequence	Intended Buffer Offset	Error Condition Code

The radio has an allocated data buffer for each option address supporting this opcode. Each data buffer also has its own data pointer. The pointer is set to the offset received in the start of the sequence Opcode. The pointer is incremented as each byte of data in the body of the

sequence is received. Because each data area has its own pointer, option broadcasts may be interleaved with those from other options.

When a long transmission is complete, XFREND may optionally be sent. This forces the destination to verify that it received all the bytes sent over the bus. The destination must then respond with an acknowledgement (Error Code 00, see below). On short transfers, normal serial bus error recovery will suffice to insure successful communications.

The intended destination device broadcasts XFRERR immediately upon detecting an error or in response to XFREND. If no errors are detected the response to XFREND would be XFRERR with a code of 00. Error Codes are defined below:

Error Condition Codes	
00	No Error Acknowledge of complete transmission.
01	Request to resend or update, no specific reason.
02	XFREND received but Pointer < Number of bytes sent.
03	Pointer > Number of Bytes to Follow.
04	Number of Bytes to Follow <> Number Just Sent.
05	Assigned Buffer size < Number of Bytes to Follow.
all remaining error codes are reserved for future use.	

To allow Emergency messages, error messages, and other bus traffic, options (and radio) should not send long sequences of data so fast that other serial bus options never have a chance to grab busy and access the bus. Many options grab busy within 1.2 mS after busy is released but, the Spectra radio can delay up to 2.5 mS after the release of busy to grab it for a pending broadcast. Options sending long messages should pause at least 2.5 mS between the last release of busy and the next attempt to pull busy to give the Spectra radio a chance of grabbing busy on the second try when it has a pending broadcast.

XTLPUL - PROCESSOR CRYSTAL PULL - opcode $54

This opcode is used by the radio processor to change the crystal shift state of options with crystal pull hardware. Processor crystal harmonics can and will quiet the radio receiver by radiating directly into the IF string or by externally radiating the mobile antenna. This opcode carries both the source and destination addresses. The source GROUP is always zero so that crystal pull information is broadcast to all groups in a multiple radio system. The destination address may be as specific or as global as required in selecting which processors pull their crystals.

Bit

7	6	5	4	3	2	1	0
0	0	0	Device Address				
x	x	x	x	x	x	x	D1
Data 2							
0	Opcode						
CRC							

D1: = 0 Release Crystal Shift.
= 1 Shift Crystal.

DATA 2: = Destination GROUP/ADDRESS

The source of this opcode is responsible for making crystal pull determinations. Usually space in the radio EEPROM is allocated for crystal pull data on a mode by mode basis. When the destination DEVICE ADDRESS = $00, then all devices in the specified GROUP will be addressed. This is useful in multi-radio systems.

Note: At 9600 bps, the serial bus may not be fast enough to Broadcast this opcode during channel scan.

4 MULTIPLE SYSTEMS

Installations exist which incorporate several radios (eg. high band, low band, and UHF) and control heads (eg. fire engines). SB9600 (then named the SYSTEMS 9000 bus or LONGHORN) was structured to allow these to all be connected into a common bus, resulting in simple installation and a greater degree of system flexibility. This actually represents a single system with multiple nodes and can be described by individually considering how multiple radios and multiple control heads can be incorporated. The ultimate installation is a console with multiple dispatchers and CRT's, all controlling various radios with varying options.

4.1 Multi-Control Head Systems

A multi-control head system involves controlling the radio system from more than 1 location. There are several consequences to this. The first is that volume controls for the 2 locations should be independent. The 'front' control head operates the normal radio volume while the 'rear' control head requires its own volume attenuator. Its audio comes from FLTAUD and is amplified with a separate audio power amplifier. Muting is based on AUDMUT sent by the radio and may be implemented by external relays. If a 'Power Voice Speaker' is used, then the volume control on the 'rear' control head has no effect.

Another consequence is that all 'rear' control heads must be smart. They must know not to send volume information to the radio and that they do not reside at addresses 5-6. They also must control relays to mute FLTAUD (and perhaps MIC).

It may be desirable to disable a 'rear' control head. In such a case, a 'rear' control head must be able to turn off the display, ignore buttons and switches (but not the bus), mute FLTAUD, and mute MIC. It must also then decode BUTTON opcodes and send an appropriate DISPLY.

While operational, the 'rear' control head is transparent to the bus. A button pressed on it will have the same effect as if it were pressed on the 'front' control head (ie. only destination information is sent with BUTTON opcodes). Also DISPLY and DSPMSG will result in the same FIELD on both control heads (ie. only source information is sent with DISPLY and DSPMSG opcodes). The only time they will show different information is when one of them is locally editing a display (see Section 3.6.1). Once the edited value is selected (SETBUT), then the responding DISPLY will cause the two control heads to again show the same value.

It is entirely possible that the two control heads be of different styles. Given the same FIELD, they show different physical displays based on their display mapping. Also, the control heads can have different options available, depending on button mapping (the FIELD resulting from the option should still result in a meaningful display even though that option is not accessible from that control head - otherwise the two operators will be operating from different displays). Simply the absence of a button is sufficient to 'delete' an option from a control head.

There does exist a potential problem which involves reading switches (eg. PTT, IGN, and HUB). These should ideally be ’wired- ORed’ to the active state. Otherwise, if PTT is pressed on both control heads and one is released, a SETBUT indicating PTT release will be sent. This is incorrect since the other PTT is still held. One solution is to physically connect the wires together. This is not always possible, however, as can be seen when considering a hand-held control head (the MIC and PTT are built into the control head - there is no PTT wire). An alternative is to have all control heads send PTT SETBUTs to one control head (located at address 5-6) and have it do the proper logic. It then sends the ’real’ PTT SETBUT to GROUP/ADDRESS = 0.

An intercom feature can also be incorporated. It functions as a special purpose option which routes the rear MIC to RXAUD, routes front MIC to the rear audio (and disconnects FLTAUD from the rear audio), sends PTTINH, and mutes audio while the user is talking. This option is fully addressable since it can be turned on/off and needs its own switching circuitry to manipulate the rear audio (the front audio can be manipulated via the bus with RXAUD). It is possible that it can be built into a ’rear’ control head itself.

An EMS feature is related to the above but deals with audio priority while transmitting. Although either control head could transmit, usually only 1 MIC audio is desirable. This option can enable/disable the front MIC (via TXAUD) or the rear MIC (via local switching) based on the assigned priority. It is even possible to sum the two if necessary, although available deviation may limit the audio. Another feature is automatic monitoring of the other control head’s transmission. This is similar to the intercom feature described above. This option, like the intercom, is not a general purpose option that can connect anywhere on the bus. It is intimately related to the ’rear’ control head and needs additional access to the ’rear’ audio lines.

Although a description of a typical dual-control head was described, with custom programming, almost any configuration is possible. The first effort will probably be to allow different displays on the two control heads. A typical example is an ambulance where the driver wants to see a SIREN page while the medical technician wants to see the EKG display.

4.2 Multi-Radio Systems

Several different implementations of multi-radio control are possible. However, they all will have some common characteristics. The majority of decoding options can only look at 1 discriminator at a time. Thus, each radio will need its own decoder, they cannot be shared. Also, the radios must be connected together via an interface box which provides RXAUD buffering, TXAUD buffering, MIC buffering (if only 1 MIC line is used), DISC summing and buffering, and speaker summing and buffering (if only 1 speaker is used).

The key in implementing multi-radio systems is the GROUP address. Each radio and its options must be assigned a non-zero GROUP. These help isolate the GROUPS from each other (eg. a data decoder will only mute the audio of the radio in its GROUP). Each GROUP will respond only to commands directed to that GROUP (or GROUP 000) as well as only generate commands to that GROUP (or sometimes GROUP 000).

Some options operate the same regardless of how many radios are in the system (SIREN is the traditional example). They are assigned GROUP 0 and listen to all activity on the bus (during XRAD, audio will unmute if AUDMUT is received from any GROUP), as well as generate commands that affect all radios (during PA, PTTINH is sent to GROUP 000 forcing all radios to disable PTT).

Certain multi-radio implementations may require 'custom' software. Although SB9600 will support multi-radio control, the human interface and certain option features are not at all trivial and will require special considerations. Can more than 1 radio transmit at the same time? What about during Talkback scan? What if a data option needs to respond on radio B while radio A is transmitting? Can all radios be heard simultaneously? What is displayed when all radios lock onto carriers during channel scan? SB9600 provides the fundamental tools needed to build such systems.

It should be noted that a Duplex radio is considered a single radio. There is only 1 RXAUD, TXAUD, MIC, DISC, etc. SB9600 does not preclude Duplex operation. Both receive and transmit operations can occur simultaneously if the hardware and software can support it (eg. RADKEY has no effect on RXAUD, encode and decode filters can be used simultaneously, etc).

5 EXAMPLES

The examples on the following pages illustrate how the various opcodes can be used by the options to control radio functions. These can be used as guidelines to help instill the spirit in which the opcodes were meant to be used. Included in each example is a description of what is taking place, a listing of the opcodes that are used, and a description of how they are used. When viewing data on the bus, the order in which the opcodes appear may not entirely match the order presented here. This is because there is a time delay between when an option wants to send something and when it actually is able to. The opcodes have been ordered so that dependent functions are grouped together. An example of this re-ordering is shown below:

ACTUAL RE-ORDERED

Let’s reconfigure Securenet ENTBUT ENTBUT
Pick the next code INCBUT INCBUT
Securenet displays the code DISPLY DISPLY
It is invalid, send ”bad beep” ALRTTN ALRTTN
We are done trying to change code EXTBUT AUDMUT
Securenet displays the active code DISPLY AUDMUT
Radio unmutes to generate tone AUDMUT EXTBUT
Radio mutes when tone is done AUDMUT DISPLY

Note that the meaning and relationship of the opcodes are more clear when grouped together according to function (ie. the AUDMUT opcodes are a result of ALRTTN). In this particular case, the radio responded more slowly on the bus than both the control head and the option. In other radio systems, the appearance of data on the bus may be different again.

5.1 Securenet

First it is desired to change to code #2. After selecting this new code. A coded transmission occurs.

ACTIVITY OPCODE DATA FROM

Reconfigure Securenet ENTBUT code CH
Sound configuration beep ALRTTN $03,$01 CH
Radio unmutes to generate tone AUDMUT $01 RADIO
Radio mutes when tone is done AUDMUT $00 RADIO
Securenet prepares to transmit TXAUD $80,$D2 DVP
Securenet ON indicator DISPLY on, $01 DVP
Ask for current code # SHOBUT code CH
Show current CODE # DISPLY code, 1 DVP
Wait for button press.
See next code INCBUT code CH
Show CODE # DISPLY code, 2 DVP
Wait for button press.

Pick the code EXTBUT code, 2 DVP
Sound exit configuration beep ALRTTN $03,$01 CH
Radio unmutes to generate tone AUDMUT $01 RADIO
Radio mutes when tone is done AUDMUT $00 RADIO
Show current CODE # DISPLY code, 2 DVP
Securenet ON indicator DISPLY on, $01 DVP
Wait for something to happen.
Press PTT to transmit SETBUT $03,$01 CH
Radio keys up RADKEY $01 RADIO
Turn on TX indicator DISPLY TX, $01 RADIO
Keep radio keyed for pending EOM TXCTRL $01 DVP
CODED TRANSMISSION indicator DISPLY TX, $01 DVP
Show current CODE # DISPLY code, 2 DVP.
Encode MIC HI audio.
Release PTT SETBUT $03,$00 CH.
Send EOM.
Dekey radio when done with EOM TXCTRL $00 DVP
Release TXAUD for others to use TXAUD $00 DVP
CLEAR TRANSMISSION indicator DISPLY TX, $00 DVP
Radio de-keys RADKEY $00 RADIO
Turn off TX indicator DISPLY TX, $00 RADIO
Securenet prepares to transmit TXAUD $80,$D2 DVP.
Wait for something to happen.

5.2 MVS

A local recording is made at the mobile and played backed back automatically. A pushbutton type control head is assumed.

ACTIVITY OPCODE DATA FROM
Push the RECORD button SETBUT record CH
Inhibit PTT from keying radio PTTINH $01 MVS
Turn on RECORD indicator DISPLY record MVS
Wait for something to happen.
Press PTT to record a message SETBUT $03,$01 CH
Mute normal RX audio RXAUD $92 MVS
Radio unmutes due to RXAUD AUDMUT $01 RADIO
Sound Record beep ALRTTN $03,$01 MVS
Wait for something to happen.
Release PTT when done with message SETBUT $03,$00 CH
Playback the message RXAUD $92 MVS.
Playback message.
Release RXAUD when playback done RXAUD $00 MVS
Wait for something to happen.
Release RECORD button SETBUT off CH
Enable PTT transmissions PTTINH $00 MVS
Turn off RECORD indicator DISPLY off MVS

5.3 EMERGENCY / PTT ID

An emergency button is pressed. This causes data steering and an Acknowledge to be received. Also the Emergency monitor feature has been enabled which keeps mobile audio muted until PTT is pressed. PTT ID is also available on this option.

ACTIVITY OPCODE DATA FROM
Emergency button pressed SETBUT $04,$01 CH
Emergency button released SETBUT $04,$00 CH
Disable TX indicator TXLTIN $02,$01 EMER
Disable normal RX audio DISMUT $01 EMER
Change frequencies ACTMDW $02,freq EMER
Radio updates option enables ACTMDU enables RADIO
Prepare for transmission TXAUD $00,$B1 EMER
Key radio TXCTRL $01 EMER
Radio keys up RADKEY $01 RADIO.
Send PSK/MSK (etc.)
De-key radio when done TXCTRL $00 EMER
Radio de-keys RADKEY $00 RADIO
Wait for Acknowledge.
Radio senses carrier SQLDET $01 RADIO
BUSY indicator DISPLY busy,1 RADIO
DOS detect DISMUT $01 EMER.
Decode valid Acknowledge.
Change frequencies back ACTMDW $00,$00 EMER
Radio updates option enables ACTMDU enables RADIO
PTT ID prepares for transmission TXAUD $00,$A9 ID
Radio loses carrier SQLDET $00 RADIO
Wait for PTT to clear muting.
PTT pressed SETBUT $03,$01 CH
Radio keys up RADKEY $01 RADIO
Allow normal audio muting DISMUT $00 EMER
Allow normal TX indicator TXLTIN $00,$00 EMER
Send Tx indicator DISPLY TX, $01 RADIO
Keep radio keyed for PTT ID TXCTRL $01 ID
PTT ID sidetone ALRTTN $83,$12 ID
Radio unmutes to generate tone AUDMUT $01 RADIO
PTT ID finished ALRTTN $00,$00 ID
Radio mutes when tone is done AUDMUT $00 RADIO
Release TXAUD for others to use TXAUD $00,$00 ID
PTT ID done with TX TXCTRL $00 ID.
Transmit MIC HI audio.
PTT released SETBUT $03,$00 CH
Radio dekeys RADKEY $00 RADIO
Turn off TX indicator DISPLY TX, $00 RADIO
PTT ID prepares for transmitting TXAUD $00,$A9 ID

Appendix A ALERT TONES

DATA 4 of the ALRTTN opcode contains the Pattern Number described below where a ’0’ in the pattern indicates an 80msec period of silence and a ’1’ indicates an 80msec period of tone. When generating a tone with D1=0 (single pattern), trailing zeros may be ignored (ie. the pattern ends at the last ’1’)

.

Pattern Number	Pattern	Pattern Number	Pattern
0	0000 0000 0000 0000	10	1111 1111 0000 0000
1	1000 0000 0000 0000	11	1111 1111 1111 0000
2	1010 0000 0000 0000	12	1111 1111 1111 1111
3	1010 1000 0000 0000	13	1111 1111 1111 1100
4	1010 1010 0000 0000	14	Single 40msec Click
5	1010 1010 1000 0000	15	Phone Ring
6	1010 1010 1010 0000	16	Dynamic Regroup Tone
7	1010 1010 1010 1000	17	Talk Permit Tone
8	1010 1010 1010 1010	18	Test Mode Beeps
9	1100 0000 0000 0000	19	Volume Set Tone
A	1100 1100 0000 0000	1A	Extended Talk Permit
B	1100 1100 1100 0000	1B	TBD
C	1100 1100 1100 1100	1C	TBD
D	1101 1011 0110 1100	1D	TBD
E	1111 0000 0000 0000	1E	TBD
F	1111 0000 1111 0000	1F	TBD

’Phone Ring’ consists of:

1) approx 14 cycles of alternating 600Hz and 900Hz, each on for 20msec.

2) approx 1.2 seconds of silence.

Note: DATA 3 of ALRTTN (frequency) is ignored

'Dynamic Regroup Tone' consists of:

1) 20msec of 1800 Hz, 20msec of 900Hz, 20msec of 300Hz

2) repeat

Note: DATA 3 of ALRTTN (frequency) is ignored

'Talk Permit Tone' consists of a 1001 0011 pattern with 20msec increments.

'Test Mode Beeps' consist of a 1010 1010 1010 1010 pattern with 40msec increments.

'Volume Set Tone' consists of approx. .3 seconds of constant tone. Minimum volume settings are ignored.

'Extended Talk Permit Tone' consist of a 0010 0100 1001 0011 pattern with 20msec increments.

Appendix B MISCELLANEOUS SYS9000 DOCUMENTS

When designing a new option for SYS9000, it may become important to design-in compatibility with existing SYS900) equipment. The following list describes what documents are available. They will typically show what opcodes are received, which ones are transmitted, and how they are used by each option. Also described are the display and button handling done by each option.

12M80952T26	Syntor X radio EEPROM
12M80952T27	MVS
12M80952T28	Basic Securenet
12M80952T52	Siren/PA
12M80952T53	Control Head EEPROM
12M80952T54	Syntor X radio/control head
12M80952T55	MDC
12M80953T87	Single Tone
	Spectra

Appendix C SYS9000 OPTION CHARACTERISTICS

The following table has been generated as an aid to designing compatibility between new and existing options. Included are the items most frequently asked. It should be remembered that default addresses may change between different installations. Care must be taken when choosing priorities since there are a limited number. Usually, if two options are similar, they will not both be used at the same time and can thus share the same priority (eg. Quick Call and MDC SelCall). The entry ”-” indicates that the option does not use that resource.

DEVICE NAME	DEFAULT ADDRESS	RXAUD PRIORITY	TXAUD PRIORITY	TXAUD BAND
Control Head Extensions	$07	-	-	-
Siren/PA	$08	-	-	-
Securenet	$09	2	2	Both
Emergency	$0A	-	6	Audio
Status/ID	$0A	-	5	Audio
Message	$0B	-	5	Audio
MDC1200 SelCall	$0B	-	5	Audio
MDC600 SelCall	$0C	-	5	Audio
MVS Emergency	$0D	-	6	Audio
MVS ID	$0D	-	5	Audio
MVS Audio	$0D	4	4	Audio
Phone	$0E	6	6	Audio
DTMF	$0F	6	6	Audio
Trunking System	$10	-	7	Both
Trunking Options	$11	-	7	Both
Vehicular Repeater	$12	6	6	Audio
SP Repeater	$12, $13	6	6	Audio
Single Tone	$14, $15	-	7	Audio
Vehicle Location	$16	-	6	Audio
KDT Terminal	$17	-	2-7	Audio
Trunked deskset	$18	5	1	Audio
Metrocom	$19	-	4	Audio
Control Host	$1A	-	-	-
Vehicular Adapters	$1B	-	-	-
available	$1C	-	-	-
available	$1D	-	-	-
available	$1E	-	-	-
available	$1F	-	-	-

Appendix D SERIAL BUS OPCODE USAGE

OPCODE		SPECTRA	SIREN	MVS	MDC 600	MDC 1200	DVP	ADV DVP	SINGL	DTMF	VRS	AVL	TRUNK	KDT
00	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
01	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
02	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
03	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
04	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
05	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
06	EPREQ	-	-	-	-	-	-	-	-	-	-	-	-	-
07	MEMADD	b	-	-	r	r	-	b	-	-	-	-	r	-
08	MEMACS	r	r	r	r	r	-	r	-	r	-	r	r	-
09	SHOBUT	-	-	-	-	-	r	r	-	r	-	-	r	-
0A	SETBUT	-	r	-	-	-	r	r	-	r	r	r	r	r
0B	INCBUT	-	r	-	-	-	r	r	-	r	r	-	r	r
0C	DECBUT	-	r	-	-	-	r	r	-	r	r	-	r	r
0D	ENTBUT	-	-	-	-	-	r	r	-	r	r	-	r	t
0E	EXTBUT	-	-	-	-	-	r	r	-	r	r	-	b	t
0F	DELBUT	-	-	-	-	-	-	-	-	r	-	-	r	-
10	RCLBUT	-	-	-	-	-	-	-	-	-	-	-	r	-
11	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
14	BUTPRS	-	-	-	r	r	-	-	-	r	-	-	-	r
15	RADRDY	t	r	r	r	r	r	r	r	r	r	r	r	r
16	OPTSTS	t	r	r	r	-	r	r	r	r	r	-	-	t
96	-----	r	t	t	t	-	t	t	t	t	t	-	-	-
17	DEVJSR	r	-	-	r	r	-	-	-	-	-	-	r	-
18	PTTINH	b	b	b	b	b	r2	r2	-	-	b	b	r2	t
19	RADKEY	t	-	r	r	r	r	r	r	r	r	r	r	r
1A	RXAUD	b	r1	b	-	-	b	b	-	t5	b	b	b	b

OPCODE		SPECTRA	SIREN	MVS	MDC 600	MDC 1200	DVP	ADV DVP	SINGL	DTMF	VRS	AVL	TRUNK	KDT
1B	TXAUD	b	-	b	b	b	b	b	b	b	b	b	b	b
1C	ALRTTN	r	-	t	t	b	b8	b8	b7	t	t	b6	b7	-
1D	AUDMUT	t	r	r	-	-	r	r	-	-	r	-	-	r
1E	SQLDET	t	-	r	r	r	r	r	-	-	r	r	-	r
1F	ACTMDU	t	-	r	r	r	r	r	r	r	r	r	r	r
20	TXMODE	t	-	r	r	-	r	r	-	-	-	-	-	r
A0	-----	-	-	-	-	-	-	-	-	-	-	-	t	-
21	RXMODE	t	-	-	-	-	r	r	-	r	-	-	-	r
A1	-----	-	-	-	-	-	-	-	-	-	-	-	t	-
22	DISMUT	r	-	b	b	b	r3	r3	-	b	b	t	b	-
23	PLDECT	t	-	-	-	-	r	r	-	-	r	-	-	r
24	RPTDIR	r	-	-	-	-	-	-	-	-	-	-	-	-
25	SGINFO	t	-	-	-	t	b	b	-	-	r	r12	b9	r
26	VOLMIN	r	-	-	-	-	-	-	-	-	-	-	-	t
27	VOLVAL	r	-	-	-	-	-	-	-	-	-	b6	-	t
28	SQLVAL	r	-	-	-	-	t4	t4	-	-	-	-	-	r
29	ACRXPL	r	-	-	-	-	-	-	-	-	-	-	-	b
2A	ACTXPL	r	-	-	-	-	-	-	-	-	-	-	-	b
2B	RXPLIN	b	-	-	b	b	b	b	-	b	-	-	-	b
2C	REVINH	r	-	-	-	-	-	-	-	-	-	-	-	-
2D	SCONOF	b	-	-	-	t	-	t	-	-	-	t	-	t
2E	TXCTRL	r	-	t	t	t	t	t	-	t	t	t	b10	t
2F	ACMODW	r	-	t	t	t	-	t	-	-	b	b	-	b
30	UNQSCN	r	-	-	-	-	-	t	-	-	-	-	-	-
31	TBONOF	r	-	-	-	-	-	-	-	-	-	-	-	-
32	DEVVAL	r	-	-	-	-	-	-	-	-	-	-	-	-
33	ACPRVL	r	-	-	-	-	-	-	-	-	-	-	-	-
34	TXLTIN	r	-	t	t	t	-	-	r	-	-	b13	b11	b
35	ETONOF	r	-	-	-	-	-	-	-	-	-	-	-	-
36	TOTVAL	r	-	-	-	-	-	-	-	-	-	-	-	t

OPCODE		SPECTRA	SIREN	MVS	MDC 600	MDC 1200	DVP	ADV DVP	SINGL	DTMF	VRS	AVL	TRUNK	KDT
37	ACTNPL	r	-	-	-	-	-	-	-	-	-	-	-	-
38	ACNPLB	r	-	-	-	-	-	-	-	-	-	-	-	-
39	ACPRI1	r	-	-	-	-	-	-	-	-	-	-	-	-
3A	ACPRI2	r	-	-	-	-	-	-	-	-	-	-	-	-
3B	PRUPST	r	t	t	t	t	t	t	t	t	b	t	b	t
3C	DISPLY	t	t	t	t	t	-	-	-	t	t	-	t	b
3D	DSPMSG	t	-	-	t	t	-	-	-	-	-	-	t	b
3E	BATTST	-	-	-	-	-	-	-	-	-	-	-	-	-
3F	CHNUMB	-	-	-	-	-	-	-	-	-	-	-	t	-
40	TSTMOD	b	-	-	-	-	-	-	-	-	-	-	r	-
41	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
42	PALIM	r	-	-	-	-	-	-	-	-	-	-	-	-
43	OSCVAL	r	-	-	-	-	-	-	-	-	-	-	-	-
44	PRTVAL	b	-	-	-	-	-	-	-	-	-	-	-	-
45	TRKOPC	t	-	-	-	-	-	-	-	-	-	r	t	-
46	TRKMDU	t	-	-	r	r	r	r	r	r	r	r	r	r
47	ALARMS	-	-	-	r	r	-	-	-	r	-	-	r	-
48	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
49	ACTSTU	-	-	-	-	r	r	r	-	-	-	-	r	-
4A	PCONOF	-	-	-	-	-	-	-	-	-	t	-	r	-
4B	DATOPC	b	-	-	-	-	-	-	-	-	-	b	b	b
	RQST	r	-	-	-	-	-	-	-	-	-	b	b	t
	GRNT	t	-	-	-	-	-	-	-	-	-	r	t	r
	PEND	b	-	-	-	-	-	-	-	-	-	r	b	b
4C	DATCHN	r	-	-	-	-	-	-	-	-	-	-	r	t
4D	SYSSTS	t	-	-	-	-	-	-	-	-	-	-	t	b
4E	TIMEUP	-	-	-	-	-	-	-	-	-	-	-	-	-
4F	METINF	-	-	-	-	-	-	-	-	-	-	-	-	-
50	XFRSTR	-	-	-	-	-	-	-	-	-	-	-	-	-
51	XFRBDY	-	-	-	-	-	-	-	-	-	-	-	-	-

OPCODE		SPECTRA	SIREN	MVS	MDC 600	MDC 1200	DVP	ADV DVP	SINGL	DTMF	VRS	AVL	TRUNK	KDT
52	XFREND	-	-	-	-	-	-	-	-	-	-	-	-	-
53	XFRERR	-	-	-	-	-	-	-	-	-	-	-	-	-
54	XTLPUL	-	-	-	-	-	-	-	-	-	-	-	-	-
55	DEFBUT	-	-	-	-	-	-	-	-	-	-	-	-	-
56	SFTPT1	-	-	-	-	-	-	-	-	-	-	-	-	-
D6	-----	-	-	-	-	-	-	-	-	-	-	-	-	-
57	BUTCTL	-	-	-	-	-	-	-	-	-	-	-	-	-
58	LUMCTL	-	-	-	-	-	-	-	-	-	-	-	-	-
59	CNFREQ	-	-	-	-	-	-	-	-	-	-	-	-	-
5A	SPAOPC	-	-	-	-	-	-	-	-	-	-	-	-	-
5B	TESTOP	-	-	-	-	-	-	-	-	-	-	-	-	-

NOTES:

-- opcode is ignored 11) reset bus if other options report NAK error

t - opcode is transmitted 12) receive low speed detect from trunking option

r - opcode is received 13) to check for presence of a control head

b - opcode is transmitted and received

1) to mute XRAD during MVS and DVP audio.

2) to dekey Securenet and Trunking

3) to mute decrypted audio

4) for REX

5) only during TX for DTMF audio feedback

6) aborts if currently busy

7) hold off if currently busy

8) Securenet encrypts after side tones

9) receive from DVP, transmit low speed detect

10) for option-originated transmissions

Appendix E ADDRESS AND BUTTON REGISTER ASSIGNMENTS

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
00	00	power on/off	-	-	-	-	-	-	-	-	-	-
00	01	HUB	-	-	-	X	-	-	-	-	-	-
00	02	ignition sense	-	-	-	X	-	-	-	-	-	-
00	03	PTT	-	-	-	X	-	-	-	-	-	-
00	04	emergency	X	-	-	X	-	-	-	-	-	-
00	05	horn ring VIP input	-	-	-	X	-	-	-	-	-	-
00	06	(reserved for VIP)	-	-	-	-	-	-	-	-	-	-
00	07	(reserved for VIP)	-	-	-	-	-	-	-	-	-	-
00	08	emergency clear	X	-	-	-	-	-	-	-	-	-
00	09	front/rear control	-	-	-	X	-	-	-	-	-	-
00	0A	multiple radio select	-	-	-	X	-	-	-	-	-	-
00	0B	alternate HUB	-	-	-	X	-	-	-	-	-	-
01	00	zone	X	X	-	X	-	-	-	-	-	-
01	01	mode	X	X	X	X	-	-	X	X	-	-
01	02	volume	X	X	X	-	-	-	-	-	-	-
01	03	squelch	X	X	X	-	-	-	X	X	-	-
01	04	repeat/direct	X	-	-	-	-	-	-	-	-	-
01	05	monitor	X	-	-	-	-	-	-	-	-	-
01	06	extender	-	-	-	-	-	-	-	-	-	-

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
01	07	RF power	-	-	-	-	-	-	-	-	-	-
01	08	home	-	-	-	X	-	-	-	-	-	-
01	09	radio speaker mute	X	-	-	-	-	-	-	-	-	-
01	0D	alarms	X	X	X	-	-	-	X	X	-	-
01	0E	alarms on/off	X	-	-	-	-	-	-	-	-	-
01	0F	alarms select	X	X	X	-	-	-	X	X	-	-
01	10	radio select	-	-	-	-	-	-	-	-	-	-
01	11	RSSI	-	-	X	-	-	-	-	-	-	-
01	12	Spkr Routing, Fixed Volume, Fixed RF Power (Refer to Jedi Vehicular Adapter Documentation)	-	-	-	X	-	-	-	-	-	-
01	30-35	Spectra Ctrl Head function buttons	X	-	-	-	-	-	-	-	-	-
01	38-3A	Spectra Ctrl Head with no DEK VIP IN 1-3 data = $00 (closed) or $01 (open)	-	-	-	X	-	-	-	-	-	-
01	40-4F	Spectra Ctrl Head keypad buttons	X	-	-	-	-	-	-	-	-	-
01	50-57	Spectra Ctrl Head DEK A buttons	-	-	-	X	-	-	-	-	-	-
01	58-5F	Spectra Ctrl Head DEK B buttons	-	-	-	X	-	-	-	-	-	-
01	60-67	Spectra Ctrl Head DEK C buttons	-	-	-	X	-	-	-	-	-	-
01	68-70	DEK VIP In (3 per DEK)	-	-	-	X	-	-	-	-	-	-
02	00	scan on/off	X	-	-	-	-	-	-	-	-	-
02	01	scan mode	X	X	X	X	X	X	X	X	-	-
02	02	dynamic scan functions	-	-	-	X	X	X	-	-	-	-

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
02	03	talkback scan	X	-	-	-	-	-	-	-	-	-
03	00	MPL on/off	X	-	-	-	-	-	-	-	-	-
03	01	MPL code	X	X	X	X	-	-	X	X	-	-
04	00	SS on/off	X	-	-	-	-	-	-	-	-	-
04	01	SS continue	X	-	-	-	-	-	-	-	-	-
04	02	Route/Run config, Metro	X	X	X	X	-	X	X	X	X	X
04	03	Employee # config, Metro	X	X	X	X	-	X	X	X	X	X
04	04	clear button, Metrocom	-	-	-	-	-	X	-	-	-	-
04	05	PRTT button, Metrocom	X	-	-	-	-	-	-	-	-	-
04	06	RTT button, Metrocom	X	-	-	-	-	-	-	-	-	-
04	07	Data config, Metrocom	X	X	X	X	-	X	X	X	X	X
04	08	Text Message Waiting	X	X	X	X	-	X	X	X	X	-
04	09	Text Message Ack	-	-	-	X	-	-	-	-	-	-
04	0A	Clock/Status, [Time]	X	-	X	-	-	-	-	-	-	-
05	--	reserved for main control head	-	-	-	-	-	-	-	-	-	-
05	03	Spectra Dim/Backlight	-	-	-	X	-	-	-	-	-	-
05	04	‘A4’ Ctl. Head Flash Display	-	-	-	X	-	-	-	-	-	-
05	06	‘A4’ Ctl. Head Rotary Volume Set	-	-	-	X	-	-	-	-	-	-
06	--	reserved for remote control head	-	-	-	-	-	-	-	-	-	-
07	0...7	Mech Alarms (8), Metro	-	-	X	-	-	-	-	-	-	-
07	8...A	Mech Alarms update (8)	-	-	X	-	-	-	-	-	-	-

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
07	10...13	Metro VIP inputs (4)	-	-	X	-	-	-	-	-	-	-
07	14...17	Metro VIP updates (4)	-	-	X	-	-	-	-	-	-	-
08	00	siren on/off	X	-	-	-	-	-	-	-	-	-
08	01	siren type(wail,yelp,..)	X	-	-	-	-	-	-	-	-	-
08	02	PA on/off	X	-	-	-	-	-	-	-	-	-
08	03	PA volume	X	X	X	-	-	-	-	-	-	-
08	04	ExRd on/off	X	-	-	-	-	-	-	-	-	-
08	06	ExRd on/off (external)	X	-	-	-	-	-	-	-	-	-
09	00	DVP on/off	X	-	-	-	-	-	-	-	-	-
09	01	DVP code	X	X	X	X	-	-	X	X	-	-
0A	01	keypad status	X	X	X	X	-	-	X	X	-	-
0A	02	DEK status	-	-	-	X	-	-	-	-	-	-
0A	15	send program req.	X	-	-	-	-	-	-	-	-	-
0A	16	repeater access	X	X	X	X	-	-	X	X	-	-
0A	17	Smart VIP #1 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0A	18	Smart VIP #2 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0A	19	Smart VIP #3 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0A	1A	Smart VIP #4 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0A	1B	Smart VIP #5 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0A	1C	Smart VIP #6 (state 0/1)	-	-	-	X	-	-	-	-	-	-
0B	01	keypad message	X	X	X	X	-	-	X	X	-	-

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
0B	02	DEK message	-	-	-	X	-	-	-	-	-	-
0B	13	call alert	X	X	X	-	-	-	X	X	X	-
0B	14	private call	X	X	X	-	-	-	X	X	X	-
0B	19	one button call alert	-	-	-	-	-	-	-	-	-	-
0C	01	destination (sel call)	X	X	X	X	-	-	X	X	-	-
0C	02	unit to unit call	-	-	-	X	-	-	-	-	X	-
0C	03	keypad H/L alarms	X	X	X	X	-	-	X	X	-	-
0C	05	ind. H/L alarms on/off	X	-	-	-	-	-	-	-	-	-
0C	06	ind. H/L alarms select	X	X	X	X	-	-	X	X	-	-
0C	07	unit-to-unit page	X	X	X	X	-	-	X	X	X	-
0C	08	prvt. page	-	-	-	-	-	-	-	-	-	-
0C	0B	rptr. select	-	-	-	-	-	-	-	-	-	-
0D	01	MVS functions	X	X	X	-	-	-	X	X	-	-
0E	01	conv. phone	X	X	X	-	-	-	X	X	X	-
0F	01	conv. dtmf	X	X	X	-	-	-	X	X	X	-
10	05	AMSS lock	X	-	X	-	-	-	-	-	-	-
10	06	AMSS site	X	-	X	-	-	-	-	-	-	-
10	07	trunking status	X	X	X	X	-	-	X	X	X	-
10	08	trunking message	X	X	X	X	-	-	X	X	X	-
10	09	DEK trunking status	-	-	-	X	-	-	-	-	-	-
10	0A	DEK trunking message	-	-	-	X	-	-	-	-	-	-

ADDRESS	REGISTER	FUNCTION	INCBUT 0B	DECBUT 0C	SHOBUT 09	SETBUT 0A	RCLBUT 10	DLTBUT 0F	ENTBUT 0D	EXTBUT 0E	BUTPRS 14	REQBUT_11
10	0E	send rprgm. req.	X	-	-	-	-	-	-	-	-	-
10	0F	pac on/off	-	-	-	X	-	-	-	-	-	-
10	10	system wide on/off	-	-	-	X	-	-	-	-	-	-
11	01	trunking phone	X	X	X	-	-	-	X	X	X	-
11	03	call alert	X	X	X	-	-	-	X	X	X	-
11	06	private call	X	X	X	-	-	-	X	X	X	-
11	07	conv. dtmf page	-	-	-	-	-	-	-	-	-	-
11	08	one button call alert	-	-	-	-	-	-	-	-	-	-
12	--	(reserved for vehicular repeater?)	-	-	-	-	-	-	-	-	-	-
13	00	penn state rpt on/off	X	-	-	-	-	-	-	-	-	-
14	00	single tone on/off	X	-	-	-	-	-	-	-	-	-
15	--	used by single tone	-	-	-	-	-	-	-	-	-	-
16	--	AVL or Metrocom Aux Location	-	-	-	-	-	-	-	-	-	-
17	--	reserved for KDT data terminals	-	-	-	-	-	-	-	-	-	-
18	--	reserved for Trunked Deskset	-	-	-	-	-	-	-	-	-	-
19	--	reserved for MDC 4800 Com Device	-	-	-	-	-	-	-	-	-	-
1A	--	reserved for Control Host Devices	-	-	-	-	-	-	-	-	-	-
	00	Digital Radio Bit Error Rate / RSSI meausurement: enable via Data2 = $01	-	-	-	X	-	-	-	-	-	-
1B	--	reserved for Vehicular Adapters	-	-	-	-	-	-	-	-	-	-

Appendix F ADDRESS AND DISPLAY FIELD ASSIGNMENTS

ADDRESS	FIELD	DATA	DISPLAY
00	00	0,1	radio power relay off,on
00	05	0,1	horn transfer relay off,on OR Metrocom emer switch test off,on
00	06	0,1	horn relay off,on OR Metrocom driver speaker off,on
00	07	0,1	light relay off, on OR Metrocom PA speaker relay off,on
00	09	0,1	speaker MUTE LED off,on
00	0A	0,1	[dual radio] LED off,on
00	0B	0,1	PAC-RT F1/F2 (emergency steering)
00	10	0,1	Metrocom host defined output 1 (off,on)
00	11	0,1	Metrocom host defined output 2 (off,on)
00	12	0,1	Metrocom host defined output 3 (off,on)
00	13	0,1	Metrocom host defined output 4 (off,on)
00	14	0,1	Covert monitor mic relay & LED off,on
00	17	0,1	GE-STAR (PTT ID) enabled, disabled
00	21	1	[radio one] LED on
00	41	1	[radio two] LED on
00	61	1	[radio three] LED on
01	01	XXX **	’ MODE XXX’ active mode
01	02	XXX	’VOLUME XXX’
01	03	XXX	’SQUELCH XXX’
01	04	0,1	RPT/DIR LED off,on
01	05	0	’MONITOR OFF’
01	05	1	’MONITOR ON ’
01	06	---	reserved for extender
01	07	---	reserved for RF power
01	08	0,1	Home LED off, on
01	09	0	NP LED off, PRI LED off
01	09	1	NP LED on, PRI LED off
01	09	2	NP LED off, PRI LED on
01	09	4	NP LED off, PRI LED blink
01	0A	0,1	BUSY LED off,on
01	0B	0,1	TX LED off,on
01	0C	XXX **	MODE XXX SCAN mode

ADDRESS	FIELD	DATA	DISPLAY
01	0D	0	HRN/LTS display off
01	0D	1	’ HORN ON ’
01	0D	2	’ LIGHTS ON ’
01	0D	3	’HRN/LTS ON ’
01	0E	0,1	H/L LED off/on
01	0F	0	HRN/LTS display off
01	0F	1	’ HORN ON ’
01	0F	2	’ LIGHTS ON ’
01	0F	3	’HRN/LTS ON ’
01	11	XXX	DEV XXX
01	12	XXX	COMP XXX
01	13	XXX	POWER XXX
01	14	XXX	CURNT XXX
01	15	XXX	REFOSC XXX
01	16	XXX	FAIL XXX
01	17	---	FAIL 999
01	18	XXX	RSSI XXX
01	25	0,1	MONitor LED off, on
01	71-79	0,1	DEK VIP Out (Up to 3 DEKs with 3 VIPs each)
02	00	0,1	SCAN LED off,on
02	01	XXX **	MODE XXX SCAN CONFIGURATION
02	03	0,1	talk back scan LED off,on
02	04	1	’BLANK LIST ’
02	04	2	’ LIST FULL ’
02	05	0	NP LED off, PRI LED off
02	05	1	NP LED on, PRI LED off
02	05	2	NP LED off, PRI LED on
02	05	4	NP LED off, PRI LED blink
03	00	0,1	MPL LED off,on
03	01	XXX **	MPL XXX
04	00	0,1	SS LED off,on
04	01	0 ##	’ RTT ’ (## Clr 3D display)
04	01	1	’ PRTT ’
04	01	2	’ RTT RCVD ’
04	01	3	’ PRTT RCVD ’

ADDRESS	FIELD	DATA	DISPLAY
04	01	4 #	’RTT NO ACK ’
04	01	5 #	’PRTT NO ACK’
04	01	6 ##	’ BAD ID ’
04	01	7 #	’ ID NO ACK ’
04	01	8	’USE HANDSET’
04	01	9	’NO TIME SET’
04	02	0	Rt/Rn LED off
04	02	1	Rt/Rn LED on, no blink
04	02	2	Rt/Rn LED on, blink on
04	03	0	Emp# LED off
04	03	1	Emp# LED on, no blink
04	03	2	Emp# LED on, blink on
04	06	0	PRTT, RTT (both) LEDs off
04	06	1	PRTT LED on, no blink; RTT LED off
04	06	2	RTT LED on, no blink; PRTT LED off
04	06	3	PRTT LED on, blink on; RTT LED off
04	06	4	RTT LED on, blink on; PRTT LED off
04	07	0	Data button LED off
04	07	1	Data button LED on, no blink
04	07	2	Data button LED on, blink on
04	08	0	Text Message LED off
04	08	1	Text Message LED on, no blink
04	08	2	Text Message LED on, blink on
04	09	1 #	’MSG WAITING’
04	09	2 ##	’ACK REQUIRD’
04	09	3	’NO MESSAGES’
04	09	4	’ ACK RCVD ’
04	09	5 ##	’ NO ACK ’
04	0A	---	Reserved for Clock, [Time] LED
04	0B	0,1	emergency Ack LED off,on
05	01	0,1	F/R rear speaker relay off,on
05	03	---	Rsrvd for control head dim/backlight set.
05	02	--- **	F/R ’ REMOTE ’ message
06	01	0,1	F/R front speaker relay off,on
08	00	0,1	SIREN LED off,on

ADDRESS	FIELD	DATA	DISPLAY
08	01	0	’ WAIL ’
08	01	1	’ YELP ’
08	01	2	’ HILO ’
08	01	3	’ MANUAL ’
08	01	4	’ EXT RADIO ’
08	01	5	’ AIR HORN ’
08	02	0,1	PA LED off,on
08	03	XXX	PA VOL XXX
08	04	0	WAIL DEK LED
08	04	1	YELP DEK LED
08	04	2	HILO DEK LED
08	04	3	MANUAL DEK LED
08	04	4	EXT RAD DEK LED
08	05	---	’SPKR SHORT ’ (SYS 9000 Siren Option)
08	05	0,1	PA speaker off, on (Metrocom Only)
09	00	0,1	SNET LED off,on
09	01	XXX **	CODE XXX
09	02	0	don’t blink BUSY LED if on
09	02	1	blink BUSY LED if on
09	03	0	don’t blink TX LED if on
09	03	1	blink TX LED if on
09	04	---	’KEY ERASED’
09	05	---	reserved for key loader
0A	01	XXX **	STATUS XXX (600 & 1200)
0A	02	1-8	DEK LEDs (turning one on turns rest off)
0A	03	---	NO ACK
0A	11	1	’STATUS RCVD’
0A	11	2 ##	’STS NO ACK ’
0A	11	3	’PLEASE WAIT’
0A	11	4	’ NO STATUS ’
0A	12	1-8	DEK status LEDs blink on/off
0A	13	0	emergency LED off
0A	13	1	emergency LED on
0A	13	2	emergency LED blink
0A	14	0 #	stop emergency blink

ADDRESS	FIELD	DATA	DISPLAY
0A	14	1 #	' EMERGENCY '(blinks)
0A	14	2 #	' NO EMERG '
0A	15	1	'DYN REG ERR'
0A	15	2	'DYN REG ON '
0A	15	3	'DYN REG OFF'
0A	15	4	'RQAT NO ACK'
0A	15	5	'REQUEST ACK'
0A	15	6	'PLEASE WAIT'
0A	15	7	'NO DYN REG '
0A	15	8	' PRGM RQST '
0A	16	1-9	' RPTR '
0B	01	XXX **	'MESSAGE XXX'
0B	02	1-8	DEK LEDs
0B	11	1	'MESAGE RCVD'
0B	11	2 ##	'MSG NO ACK '
0B	11	3	'PLEASE WAIT'
0B	11	4	'NO MESSAGE '
0B	12	1-8	DEK message LEDs blink on/off
0B	13	1 ##	' PAGE '
0B	13	2	' ID PAGED '
0B	13	3	'PAGE NO ACK'
0B	13	4	' STORE ID '
0B	13	5	'PLEASE WAIT'
0B	13	6	'THIS UNIT '
0B	13	7	'BAD ID '
0B	14	1 ##	' CALL '
0B	14	2	'ID - RCVD '
0B	14	3	'ID - SPRVSR'
0B	14	4	' ID CALLED '
0B	14	5	'CALL NO ACK'
0B	14	6	'SCRATCH PAD'
0B	14	7	' STORE ID '
0B	14	8	'PLEASE WAIT'
0B	14	9	'THIS UNIT '
0B	14	A	' UNIT BUSY '

ADDRESS	FIELD	DATA	DISPLAY
0B	14	B	’ NO ANSWER ’
0B	14	C	’ BAD ID ’
0B	14	D	’FLEET-WIDE ’
0B	14	E	’GROUP-WIDE ’
0B	14	F	’ CANCELLED ’
0B	15	1-9 #	UNIT XXX
0B	16	1	’FLEET WIDE ’
0B	16	2	’GROUP WIDE ’
0B	17	1	call alert LED blinks
0B	17	2	private call LED blinks
0C	01	0	’ BASE ’
0C	01	1	’ FLEET ’
0C	01	2	’ GROUP ’
0C	01	3	’ UNIT ’
0C	02	XXX **	unit names with data
0C	03	0	’HRN/LTS OFF’ (keypad)
0C	03	1	’ HORN ON ’ (keypad)
0C	03	2	’LIGHTS ON ’ (keypad)
0C	03	3	’HRN/LTS ON ’ (keypad)
0C	04	0	’ CALL BASE ’ off
0C	04	1	’ CALL BASE ’ on
0C	05	0	H/L LED off (indicator)
0C	05	1	H/L LED on (indicator)
0C	06	0	HRN/LTS off (indicator)
0C	06	1	HORN ON (indicator)
0C	06	2	LIGHTS ON (indicator)
0C	06	3	HRN/LTS ON (indicator)
0C	07	---	unit to unit page
0C	08	xxx **	unit names with data
0C	09	0,1	PVT LED off,on
0C	0A	---	PVT
0C	0B	xxx	repeater names with data
0C	0C	0,1	rptr led off,on
0D	01	0	’ PLAY ’
0D	01	1	’ REPLAY ’

ADDRESS	FIELD	DATA	DISPLAY
0D	01	2	’ RECORD ’
0D	01	3	’ SEND ’
0D	01	4	’ OFF ’
0D	02	---	’ VOICE MSG ’
0E	01	---	’SCRATCHPAD ’
0E	02	1-9 **	’PHONE XXX ’
0E	03	---	’PHONE CALL ’
0E	04	---	’STORE PHONE’
0F	01	---	’SCRATCH PAD’
0F	02	1-9 **	UNIT XXX X
0F	03	---	’ CALL ’
0F	04	1	’STORE CALL ’
10 (16)	02	1	’ FAILSOFT ’
10	02	2	’OUT OF RNGE’
10	03	0,1	Emer led off,on
10	04	1	’ EMERGENCY ’ (blinks)
10	04	2	’NO EMERGNCY’
10	05	1	’SITE LOCKED’
10	05	2	’SITE UNLCKED’ ?
10	06	XXX **	’SITE XXX ’
10	07	0	’STATUS - ’
10	07	1-8 **	’STATUS X ’
10	08	1-128 **	’MESSAGE X ’
10	09	1	’STATUS RCVD’
10	09	2	’MESSGE RCVD’
10	09	3	’ NO STATUS ’
10	09	4	’PLEASE WAIT’
10	09	5	’STS NO ACK ’
10	09	6	’MSG NO ACK ’
10	0A	1-8	DEK Status LEDs on/off
10	0B	1-8	DEK Message LEDs on/off
10	0C	1-8	DEK Status LEDs Blink on/off
10	0D	1-8	DEK Message LEDs Blink on/off
10	0E	1	’DYN REG ERR’
10	0E	2	’NO DYN REG ’

ADDRESS	FIELD	DATA	DISPLAY
10	0E	3	’DYN REG ON ’
10	0E	4	’DYN REG OFF’
10	0E	5	’RQST FAILED’
10	0E	6	’ RQST SENT ’
10	0E	7	’ RPGM RQST ’
10	0F	1	’ NOT AUTH ’
10	10	0,1	SYSTEM WIDE LED off,on
10	11	1-X	DATA DISPLAYS
11 (17)	01	1	’PLEASE WAIT’
11	01	2	’PHONE BUSY ’
11	01	3	’OK TO DIAL ’
11	01	4	’ERROR 10/02’
11	01	5	’KEYPAD DIAL’
11	01	6	’PHONE CALL ’
11	01	7	’NO SYSTEM ’
11	01	8	’SCRATCH PAD’
11	01	9	’STORE PHONE’
11	02	1-9 **	’ PHONE X ’
11	03	1	’ID ALERTED ’
11	03	2	’NO ID ACK ’
11	03	3	’STORE ID ’
11	03	4	’NO SYSTEM ’
11	03	5	’PLEASE WAIT’
11	03	6	’THIS UNIT ’
11	03	7	’ILLEGAL ID ’
11	03	8	’ERROR 10/02’
11	03	9	’ID-________’
11	04	1-9 **	’UNIT ID X ’
11	05	1	’ PAGE ’ (blinks)
11	05	2	’ CALL ’ (blinks)
11	06	1	’ BAD ID ’
11	06	2	’ STORE ID ’
11	06	3	’ ID - RCVD ’
11	06	4	’ID - SPRVSR’
11	06	5	’ ID - SENT ’

ADDRESS	FIELD	DATA	DISPLAY
11	06	6	’ NO ANSWER ’
11	06	7	’ EXIT ’
11	06	8	?
11	06	9 *	’ CALL ’
14 (20)	00	0,1	Single Tone LED off,on
14	01	1-16	Single Tone Mode
14	02	1-16	Single Tone DEK LED

NOTES:

* TENTATIVE ASSIGNMENT

** NAME MAY BE SUBSTITUTED BY USER

DISPLAY MESSAGE OPCODE $3D

DISPLAY MESSAGE OPCODE $3D OR DISPLAY OPCODE $3C

Appendix G TRUNKING SUB-OPCODES (OPCODE $45)

DATA 1	DATA 2	FUNCTION
0,1	01	BTXINH - TRUNKED TX INHIBIT BROADCAST SUBOPCODE Trunked option card uses this to tell radio to inhibit user and other option transmissions until a voice channel is found.
0,1	02	BPACTRL - TRUNKED PA CONTROL SUBOPCODE Trunked option card uses this to tell radio to keyup for an ISW transmission. This subopcode overrides the above subopcode.
0,1	03	BSLRDIN - TRUNKED SELECT RADIO INHIBIT SUBOPCODE Trunked option card uses this to tell the radio that it is radio inhibited.
mode #	04	BDRMODE - TRUNKED DYNAMIC REGROUPING SUBOPCODE Trunked option card uses this to steer the radio to a dynamic regrouped mode.
0,1	05	BMDSLIN - TRUNKED USER MODE SELECT INHIBIT Trunked option card uses this to tell the radio that it is mode select disabled.
-	06	BA2SRCH - TRUNKED A2 SEARCH SUBOPCODE Trunked option card uses this to tell the radio to search for the next system in the System Search and Lock list.
0,1	07	BSLDS - TRUNKED SELECT DISABLE SUBOPCODE Trunked option card uses this to tell the radio that it is select disabled during dynamic regrouping.
-	08	BA2LB - TRUNKED A2 LOOK BACK TIMER EXPIRED Trunked option card uses this to tell the radio to execute a System Search and Lock look back.
-	09	BTQUAL - TRUNKED MODE SYSTEM QUALIFICATION FLAG Trunked option card uses this to tell the radio that it has found a trunked system for System Search and Lock.
-	0A	BA2SF - TRUNKED A2 SUBFLEET SEARCH FLAG Trunked option card uses this to tell the radio to go to the lowest position subfleet in the personality for System Search and Lock. This occurs when System Search and Lock goes to a dynamic regrouping mode and it is not dynamic regrouped.
-	0B	BAVLLOC - TRUNKED AVL LOCATION SUBOPCODE Trunked option card used this to tell the AVL option that it has received a type I PC II call with target and source IDs that match this radio’s ID. The AVL option will then send location on the voice channel.
-	0C	BPHNMOD - TRUNKED PHONE MODE SUBOPCODE Trunked option card uses this to tell the AVL option that a phone call is in progress and not to interrupt by sending in location.

Appendix H DATA SUB-OPCODES

DATA MODE:

DATA 1: = $00

DATA 2: = 0 When data option in voice mode

= 1 When data option in data mode

DATA 3: = $01

This data subopcode is sent by the data option to inform the radio and options of the mode state of the radio system. A data option may put the radio into data mode only if the radio is already in voice mode. Furthermore, an option may not send the radio into voice mode unless it was the option that sent the radio to data mode. In data mode a radio timer will count down when the radio returns to the control channel from a voice channel. Before the timer expires, the radio will only grant high priority data channel requests. After the timer expires, the radio will accept low priority data channel requests made by the data option. This ensures that voice communication can occur effectively when in data mode. In data mode the radio will also guarantee that the same data channel will be accessed when returning from a voice channel request. Upon reset, the radio and options will be in voice mode.

DATA CHANNEL REQUEST:

DATA 1: = $00

DATA 2: Defined as follows:

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	D4	D3	D2		D1		

D1: = 000; Data/voice channel released

= 001; undefined priority level

= 010; non-priority wo/tx Data/voice channel request

= 011; undefined priority level

= 100; non-priority w/tx Data/voice channel request

= 101; undefined priority level

= 110; priority wo/tx Data/voice channel request

= 111; priority w/tx Data/voice channel request

D2: = 00; Data/Voice channel release request

= 01; Data channel request

= 10; Data voice channel request (for units using voice channels to transmit data)

= 11; unused

D3 = 0; Group channel request (reserved for future use)

= 1; Individual channel request

D4 = 0; No tone mute request

= 1; Tone mute request (trunked radios will mute busy, call back, talk permit, and talk prohibit tones while channel grant is active)

DATA 3: = $02

This Data subopcode is sent by the data option to inform the radio and options of a priority or non-priority data or data voice channel request. This subopcode is followed by a Data Channel Grant Subopcode ($03) that grants or denies the requested state. The priority channel requests are intended to be used in emergency situations only and should be momentary. Any option using this state excessively runs the risk of being incompatible with other data options.

Data options should monitor the serial bus for this opcode so that they do not make a request while a higher priority request is active. Data options should also monitor the bus for this opcode to see if they have been preempted by a higher priority data channel request. Also, reset, ACTMDU, or TRKMDU clear all pending requests and grants in the radio and options.

DATA CHANNEL GRANT:

DATA 1: Defined as follows:

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	X	D7	X	X	X	D6

D6 = 0; Grant not pending
= 1; Grant pending

D7 = 0; Not end of data channel list / voice channel
= 1; End of data channel list marker

DATA 2 Defined as follows:

Bit

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D5	D4	D3	D2		D1		

D1 = 000; Data/voice channel released
= 001; undefined priority level
= 010; non-priority wo/tx Data/voice channel acknowledge
= 011; undefined priority level
= 100; non-priority w/tx Data/voice channel acknowledge
= 101; undefined priority level
= 110; priority wo/tx Data/voice channel acknowledge
= 111; priority w/tx Data/voice channel acknowledge

D2 = 00; Data/Voice data release acknowledge
= 01; Data channel acknowledge
= 10; Data voice channel request (for units using voice channels to transmit data)
= 11; unused

D3 = 0; Group channel acknowledge (reserved for future use)
= 1; Individual channel acknowledge

D4 = 0; No tone mute acknowledge
= 1; Tone mute acknowledge (trunked radios will mute busy, call back, talk permit, and talk prohibit tones while channel grant is active)

D5 = 0; Data channel not granted
= 1; Data channel granted

DATA 3: = $03

This Data subopcode is sent by the radio to acknowledge a request made by the data unit for a voice or data channel. This subopcode grants or denies access to the channel and also acknowledges whether the type of data channel is data or voice. The Data Channel Grant Subopcode normally follows a Data Channel Request Subopcode ($02), a Voice Message Pending Subopcode ($04), or a Change Data Channel Request Subopcode ($06). Data options need to monitor this opcode in case the channel is released (or taken away) due to emergency or other channel pre-empting situations. The priority channel requests are intended to be used in emergency situations only and should be momentary. Any option using this state excessively runs the risk of being incompatible with other data options. If the channel is released by the radio, then this opcode will be sent with all data fields equal to zero.

The radio will grant the request (D5=1, D6=0) if it has a data channel in its list or was able to successfully request a channel through an ISW. The radio will deny the request (D5=0, D6=0) if a higher priority voice exchange (transmit or receive) or no data channels can be obtained. When an option is denied, D1=0. The radio grant may be delayed when the radio has made an ISW request for a channel and an OSW data channel busy was received, or if the ‘change data channel request’ was received with the update list indication. For either case, the radio will pend the request (D5=0, D6=1) until the radio can grant a channel or must deny the request.

Reset, ACTMDU, or TRKMDU clear all pending requests and grants in the radio and options.

VOICE MESSAGE PENDING:

DATA 1: = SOURCE DEVICE ADDRESS

DATA 2: = 0; Non-priority voice message pending
= 1; Priority voice message pending

DATA 3: = $04

This Data subopcode is sent by the data option to inform the radio that a voice message is pending. If a priority voice message opcode is sent, the opcode will be followed by a Data Channel Grant Subopcode ($03) from the radio informing everyone that the data channel is being released. For non-priority voice message opcodes, the radio will not release the channel but instead leave the choice of answering the voice call up to the operator.

Upon releasing the data channel for a priority voice message, a radio timer will count down when on the trunked control channel. While this timer is active, low priority data channel requests will be denied. This is similar to the data mode timer that runs when returning from a voice channel.

Reset, ACTMDU, or TRKMDU clear all pending voice message opcodes in the radio and options.

DATA MESSAGE PENDING:

DATA 1: = DESTINATION DEVICE ADDRESS

DATA 2: = 0; Data non-priority data message pending
= 1; Data priority data message pending

DATA 3: = $05

This Data subopcode is sent by the radio to inform the data option that a data message is pending. If a priority data message opcode is sent, the opcode will be followed by a Data Channel Request Subopcode ($02) from the data option and a Data Channel Grant Subopcode ($03) from the radio. For non-priority data message opcodes, the data option will request a data channel if the operator chooses to do so.

Reset, ACTMDU, or TRKMDU clear all pending data message opcodes in the radio and options.

CHANGE DATA CHANNEL

DATA 1: = $00

DATA 2: Defined as follows:

Bit

7	6	5	4	3	2	1	0
X	X	X	X	X	X	D2	D1

D1 = 0; Change data channel without data channel request ISW
= 1; Change data channel with data channel request ISW

D2 = 0; Change channel without updating list
= 1; Change channel after updating list

DATA 3: = $06

This Data subopcode is sent by the data option to inform the radio that the next data channel in its list should be accessed. This subopcode is used in situations where the data option is not satisfied with the present data channel (may be heavily loaded).

This subopcode is usually followed by a Data Channel Grant Subopcode ($03). If D1=1, the radio will go to the control channel and make a request for a new data channel. In order to avoid high ISW traffic on the control channel, the radio may deny the data change data channel request if the data option makes rapid requests with the ISW request option. If D2=1, the radio will idle on the control channel for several seconds to update its data channel list before granting a new data channel.

Reset, ACTMDU, or TRKMDU clear all change data channel requests in the radio and options.

Appendix I Serial Bus Expanded Protocol (SBEP)

I.1 Scope

This document defines the Serial Bus Expanded Protocol (referred to as SBEP from now on). It includes the handshaking, message format, timing and error recovery. Several opcodes are defined in order to achieve on board reprogramming of Flash EEPROM devices. This document does not define the sequences of Requests that a host must make in order to correctly reprogram a radio with Flash EEPROM. Likewise, the address fields in the Requests/Replies that contain an address field are not defined here. This is left up to each particular product to define. No baud rate recommendations are made. Each implementation is responsible for choosing a suitable baud rate.

I.2 Overview

SBEP is an addition to the SB9600 serial bus protocol that will be used to achieve high data transfer rates between options on the SB9600 bus or between an external computer and the main processor of the radio or it's options. It is intended for one to one communications between nodes on a serial bus or from the host to options that may or may not be connected. The messages consist of Requests, Broadcasts, Replies, Acknowledges (ACK) and Negative Acknowledges (NAK). Requests require a target to be connected, Broadcasts do not. Replies are only sent in response to a Request.

The protocol achieves high throughputs by allowing a particular implementation to select what baud rate to use. Up to 38.4 K baud is supported, but may be increased in the future for faster platforms. The protocol itself specifies most timing in terms of SCI byte times. Also contributing to the high throughputs is the variable byte count in SBEP messages and single byte acknowledges.

SBEP will be used for, but is not limited to, codeplug reads and writes, automatic system testing and Flash EEPROM programming. As other uses are found for this protocol, more opcodes can be defined.

When a radio is operating with the SB9600 protocol, it can be told to switch to SBEP. The device that initiates the switch to SBEP (referred to as the 'host') will send an SB9600 message (EPREQ opcode $06) to another device on the bus. The device targeted by the EPREQ message's group/address (referred to as the 'target') is requested to switch to SBEP. The target is not required to be present. During this situation, only broadcast messages make sense.

In addition to being able to enter this protocol from SB9600, it can be entered directly via bootstrap mode. Next generation open architecture radios will allow the radio processor to reprogram it's firmware device (flash or EPROM) via SBEP messages over the serial bus. Since the firmware device is likely to be blank when the radio is manufactured, this protocol in conjunction with algorithms written for bootstrap mode of the processor will allow the radio processor to write firmware into its program space device.

For broadcast the target need not be connected.The SBEP has been designed to work in a half_duplex, no hardware handshake environment. The SB9600 busy line is shown in some of the diagrams, but is not used for implementing the protocol, other than for preventing other devices on the bus from generating SB9600 bus activity while SBEP is active.

The protocol is designed so that writes of data to the target can be done efficiently. When programming target memory it is expected that the download baud rate or the memory device program time is the limiting factor to the overall throughput.

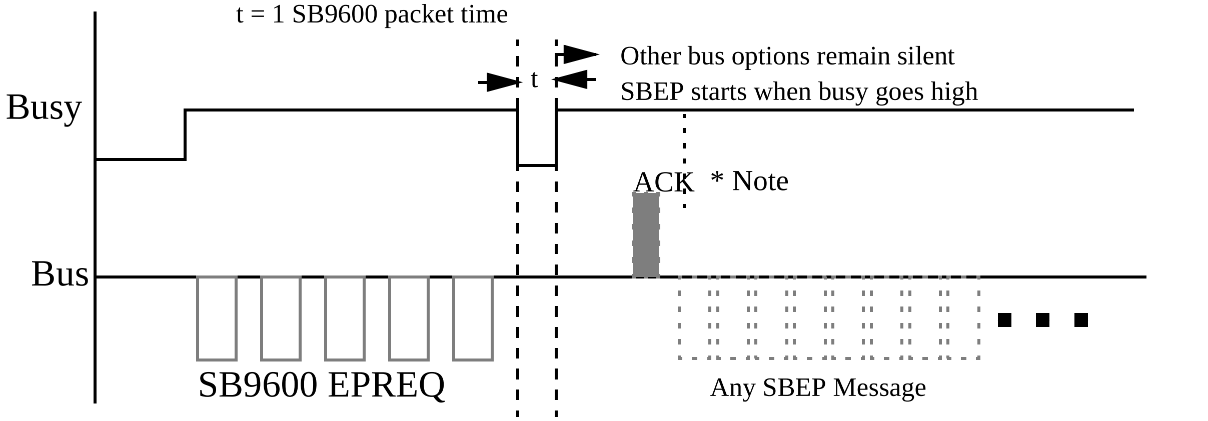
Provisions have been made in this protocol to use a radio microprocessor’s bootstrap mode for programming. This is primarily useful when the radio’s firmware needs to be changed.

Note: The bootstrap portions of the following protocol description assume that the the radio processor is a 68HC11. This is done only to illustrate and to give examples. SBEP does not specifically require a particular processor, nor does it specify a particular method of entering or exiting bootstrap. SBEP does, on the other hand, specify the amount of time that the radio is expected to be able to respond to bus activity in the context of entering or exiting bootstrap operation.

I.3 Entry to SBEP Mode

When the processor in the target has firmware in it’s EPROM, it will be operating in SB9600 mode. To change over into SBEP mode, there is an SB9600 opcode (EPREQ) that will instruct the target that the host wants to begin SBEP operation. The host will send the EPREQ SB9600 bus message, and if the target receives it and does not generate an SB9600 NAK, then the host must pull ‘busy’ active (high). All other bus options will remain quiet after seeing the EPREQ opcode and SBEP can proceed as long as ‘busy’ remains high. The timing for this method of entry into SBEP mode is illustrated by Fig Figure I-1. If the target is not present, entry to SBEP will proceed as if it was present.

Upon entry to SBEP when a target is connected, it (the target) must ACK at the SBEP baud rate (specified in the EPREQ message) as soon as it is ready. The host should wait for this ACK for 5 SCI byte times at the SBEP baud rate, and if no ACK is seen after 5 SCI byte times the target may not be present. This initial ACK is used to tell the host that it may send a message right away rather than wait for the 5 SCI byte times. If the target is late to ACK and the host proceeds to send a Request or Broadcast, there could be a collision. If such a collision occurs, the target will NAK the Request or Broadcast. The host is responsible for retrying.



* Note: Host may begin sending SBEP messages once ACK is received or after 5 SCI byte times if no ACK is seen

Figure I-1 Entry into SBEP mode via EPREQ message

If the target has no firmware to begin with, as is the case for a new controller board, then the host cannot expect the target to be in SBEP mode after issuing the EPREQ message. The EPREQ message must still be issued so that other bus options are aware that SBEP is about to occur on the bus. Since the target is unreachable, the only means of communicating with the target is to put the target in bootstrap mode and to download bootstrap code to it. The target (if it is an HC11) must come out of reset with the MODA/B lines low in order to power up bootstrap mode. It is the responsibility of the host to insure that MODA/B are low when the target resets. The next generation radio interface box and radio architecture will allow external control over the required lines for this to occur. Figure I-2 illustrates the events necessary to put the processor in bootstrap mode.

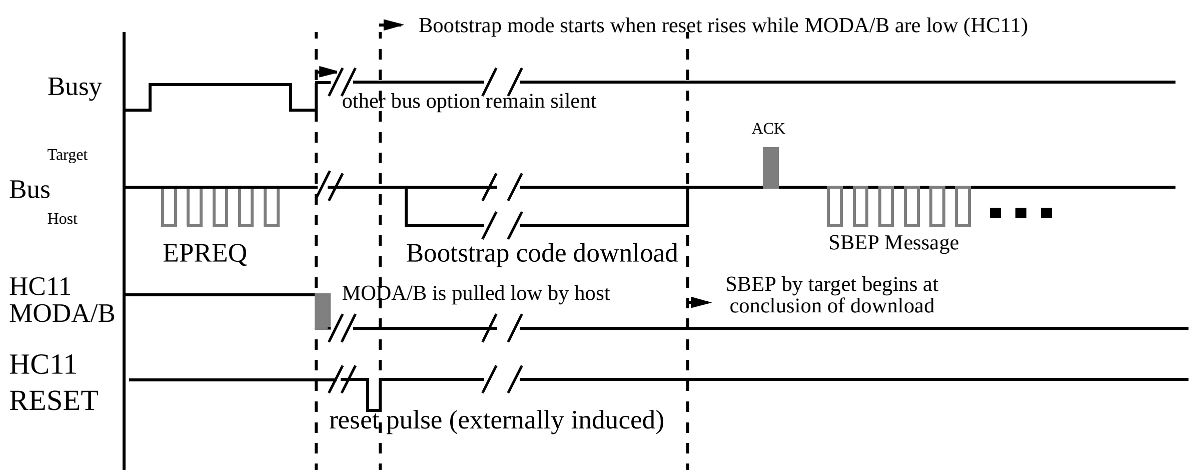


Figure I-2 Entry to SBEP when target has no firmware (HC11)

The reset pulse shown in Figure I-2 cannot be generated through the use of an SB9600 message or SBEP message as the target cannot be instructed over the serial bus to reset. In a factory environment, this reset will be induced by automatically removing power from the target for enough time to cause a reset. In the field, however, this reset must be manually performed by the person operating the reprogramming equipment. Under normal circumstances, there will be few cases in the field when the operator must manually cause the target to reset.

Note that the protocol for downloading the bootstrap code is dependent on the processor and it's clock speed. Exactly how this code is downloaded is not addressed by the SBEP protocol definition. Refer to documentation on the specific target processor for more information on this.

The last case to consider for entry into SBEP is when the target has firmware to communicate via SB9600, but it must be put into bootstrap mode. This is the case when the firmware of the target is to be upgraded. The procedure is illustrated in Fig. . The target is put into SBEP mode similar to Fig. Figure I-1, but soon thereafter an SBEP RESET message is issued to the target. The MODA/B lines are set low prior to issuing the SBEP reset so that the target (HC11) resets in bootstrap mode.

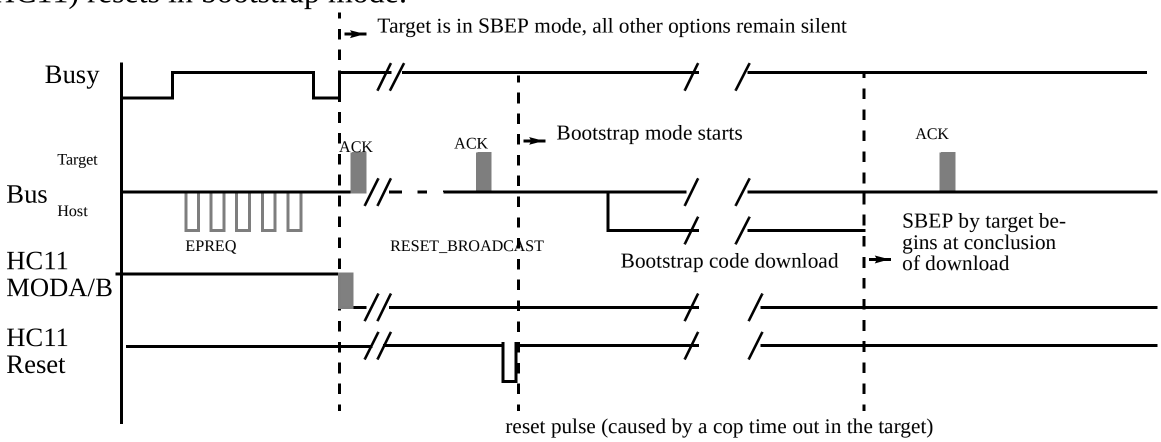


Figure I-3 Entry to Bootstrap mode via SBEP from SB9600 (HC11)

I.4 Exit from SBEP mode

When SBEP was entered from SB9600 and the target was not put into bootstrap mode, exit from SBEP can be achieved by deactivating the busy line. Once busy is released, the target processor must return to SB9600. This is illustrated in Figure I-4. All other bus options will observe 'busy' going false (low) and reactivate themselves in the SB9600 protocol.

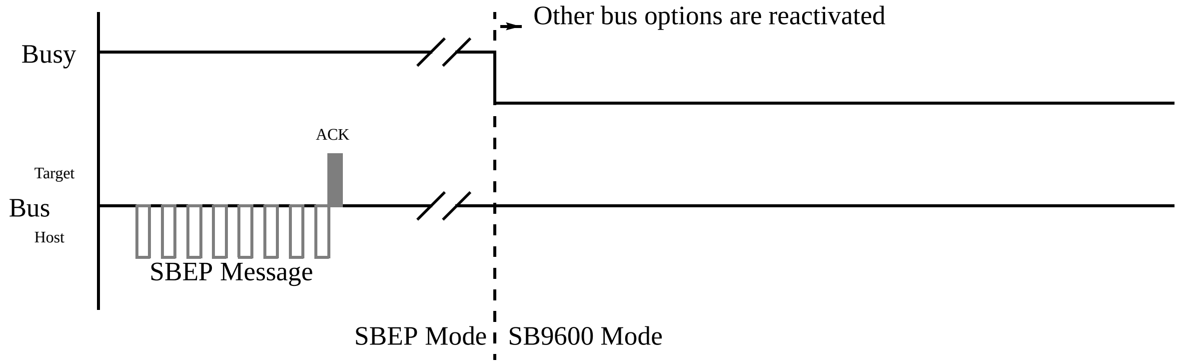


Figure I-4 Exit from SBEP, non-bootstrap mode

Alternatively, the host may take the target out of SBEP by issuing it a message. The RESET_BROADCAST message will cause the target to go through a reset and thereby power up operating normal SB9600 bus protocol. Exit via a reset is illustrated in Figure I-5.

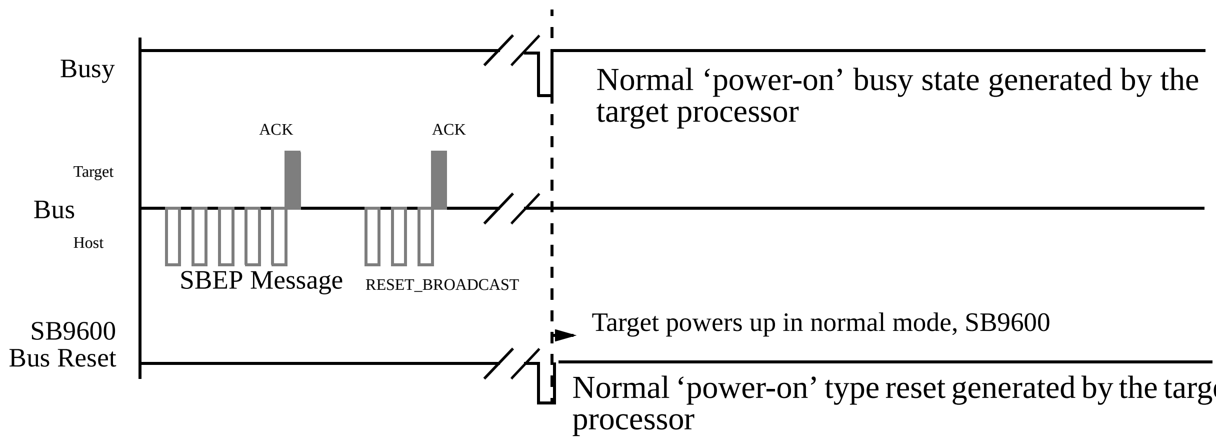


Figure I-5 Exit from SBEP, non-bootstrap mode using RESET_BROADCAST

Alternatively, if the target was in bootstrap mode getting it's firmware programmed, an SBEP RESET_BROADCAST message must be sent to the target, and the condition that allowed it to go into bootstrap mode originally must be removed (MODA/B set low). Busy must also be released. If the target has firmware to execute, then it will come up running SB9600 mode. This is illustrated in Figure I-6.

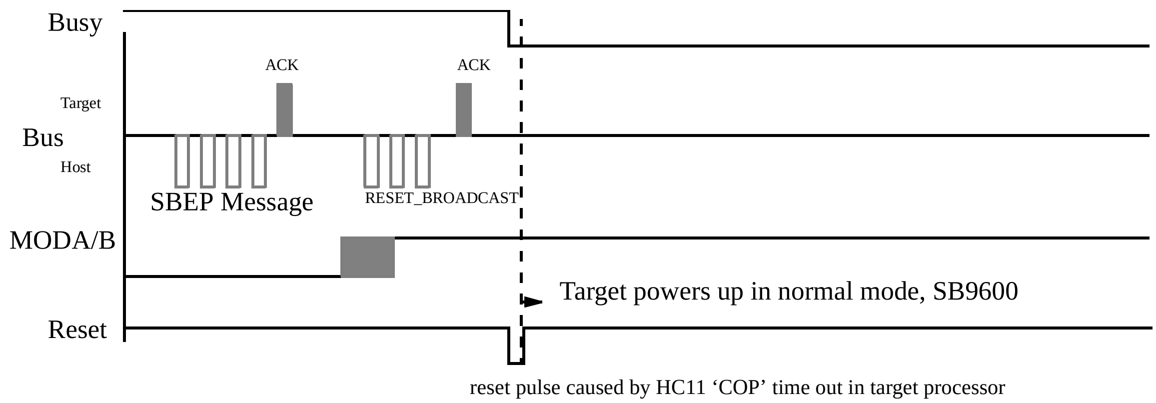


Figure I-6 Exit from SBEP and bootstrap mode using RESET_BROADCAST

I.5 Bus Protocol Definitions

Message Format

The format of SBEP messages is very flexible in that messages can be of variable length starting from one byte up to 2^16+4 bytes long where the checksum is included in the term 2^16. Practically speaking, the largest message length will be limited by the amount of RAM available in the target.

The first byte of the message always determines what is to follow. Table I-1 illustrates the possibilities for the first byte and how it affects what subsequent bytes mean.

The first byte of the SBEP message is to be considered on a per nibble basis. That is, the first nibble or most significant nibble (msn) contains opcode information and the second nibble or least significant nibble (lsn) contains information pertaining to the number of bytes to follow. Therefore, if the lsn is $0, there are no more bytes to follow, and all initial bytes that have $0 as the lsn are one byte messages. ACK and NAK are one byte messages and have no checksum.

The lsn of the first byte can take on values from $0-$E or $F. When the lsn is $0-$E it represents the number of bytes to follow in the particular message. When it is $F, it signifies that two additional bytes of extended size follow and they contain the message size.

The msn of the first byte is the opcode. It can take on values from $0-$E or $F. When it is $0-$E, it is the opcode. If it is $F, then there is an additional byte to follow that is the extended opcode.

	msn	lsn	Implication to rest of message
1	$0-$E	$0-$E	No extensions, count to checksum in lsn
2	$0-$E	$F	Two extended size bytes to follow
3	$F	$0-$E	One extended opcode byte to follow, count to checksum in lsn
4	$F	$F	One extended opcode byte, two extended size bytes to follow

Note that the extended opcode is NOT contained in the lsn count.

Table I.1 First byte of SBEP message

Combination three contains an $F in the msn, indicating that there is an additional opcode byte to follow. The extended opcode is not part of the count in the lsn as the extended opcode is known to be there by virtue of the fact that the msn was $F.

Except for single byte messages there is always a checksum as the last byte of a message. The checksum is calculated as follows:

total = sum of all bytes in the message except the checksum byte

checksum = $FF-(total mod 256)

Figure I-7 illustrates some sample messages of various sizes that are possible with this message format. The numerical values in the opcode fields are used in these examples solely for illustration purposes. It should not be inferred that any opcodes shown here have been defined as shown. (Please refer to Table and the “Extended Opcodes” section for actual opcode definitions).

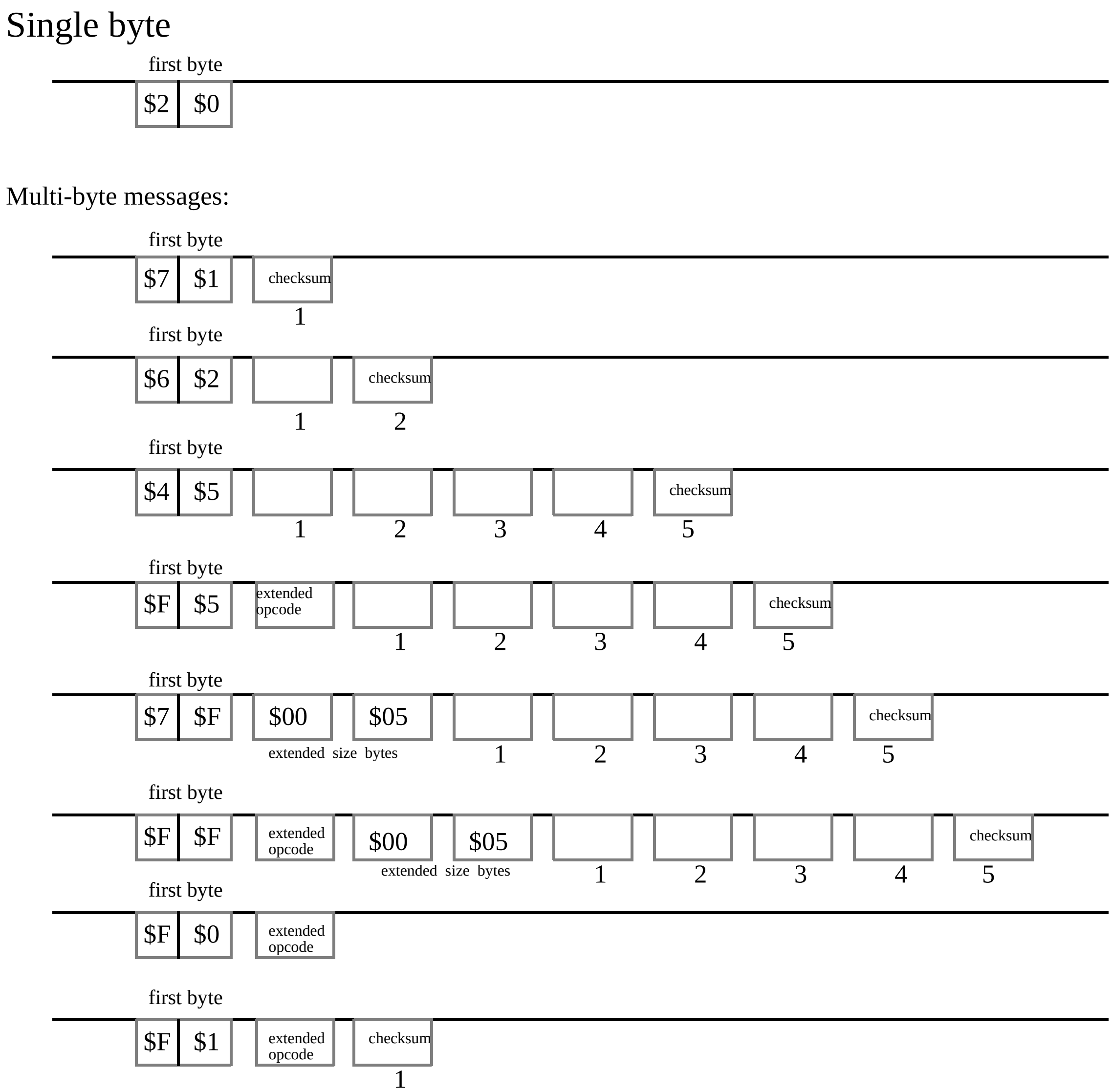


Figure I-7 Sample SBEP messages of different sizes

I.5.1 Handshaking

SBEP protocol requires that there never be more than one host and one target. The host is defined to be the initiator of SBEP, i.e. the device that initially sent the SB9600 EPREQ message. The target is the device who's address was in the initial SB9600 EPREQ message. The radio can be the host or the target for a session of SBEP. Host messages are classified into two categories based on what kind of acknowledges are required. They are Broadcasts and Requests. The target can only generate ACKs, NAKs in response to Broadcasts from the host, and ACKs, NAKs or Replies in response to Requests from the host. For the duration of an SBEP session (the entire time busy is held active) the host remains in control and must initiate any messages. The target remains the 'slave' and only responds to messages sent by the host. A discussion of each message type follows.

Broadcasts

Broadcasts are initiated from the host and do not require an ACK or a NAK, but will accept one if it is present. They are intended to be used when the radio initiates SBEP as a host to inform the outside world of events occurring in the radio. Since a target may or may not be connected, an ACK or NAK is not required for a Broadcast. A host that initiates a Broadcast must be aware that ACKs/NAKs are optional

If there is a target connected, it must ACK or NAK a broadcast. If the target ACKs then the host may send another message if it so desires. If the target NAKs, then there is positive confirmation that the Broadcast did not get through, so the host must retry up to four times until an ACK is seen or no ACK/NAK is seen (case when the target was removed half way through a retry).

If there is no target connected, then the host will never see an ACK or a NAK. Since the host does not know if there is a target connected after sending the first broadcast, it must wait 10 SCI byte times to see if an ACK or a NAK appear on the bus. If the host does not see anything in 10 SCI byte times, it can assume no target is connected, and proceed to send another broadcast. Obviously, the host should never send a Request after realizing that there is no target connected. If it does, it will go through a retry sequence and never see an ACK/NAK or reply.

If the target does ACK a broadcast, that is all it must do. There are no Replies to Broadcasts. Figure I-8 illustrates Broadcasts.

Requests

Requests are initiated by the host and require an ACK or a NAK. If the host does not see either, it must conclude that no target is connected, and that Requests no longer make sense. This would be the case if the target was removed half way through a SBEP session.

Requests always require some sort of Reply. They will be sent to the host from the target after the ACK. The host is not allowed to send another Request or Broadcast until the Reply for the last Request is seen.

I.5.2 ACK/NAK

ACKs and NAKs are sent by the target to the host to any message (Broadcast or Request) seen.

ACKs can be sent immediately by the target once a correct message checksum is received. An ACK serves to tell the host that the message (Broadcast or Request) was received correctly by the radio, and that the host should not retry. ACKs are only sent from the target to the host, never from the host to the target. See Replies.

ACKs must be sent within a 10 SCI byte time window after the checksum is received by the target. Typically, a host will be able to send this ACK almost immediately.

NAKs are sent by the target to the to the host when a corrupted message is receive by the target. A bad checksum or incorrect count of bytes will cause the target to perceive a bad

checksum and set up to send a NAK. A NAK cannot be sent immediately like the ACK because the bus may be busy sending more bytes even though the target already decided that it was going to NAK (as would be the case if the count byte is corrupted).

NAKs must be sent 5 SCI byte times after the bus has gone idle. That is, the host could continue sending a message, but the target must wait until the bus is idle for 5 SCI byte times. The NAK must be sent before 10 SCI byte times from the last byte go by. This time limit is required so that the host knows when to retry.

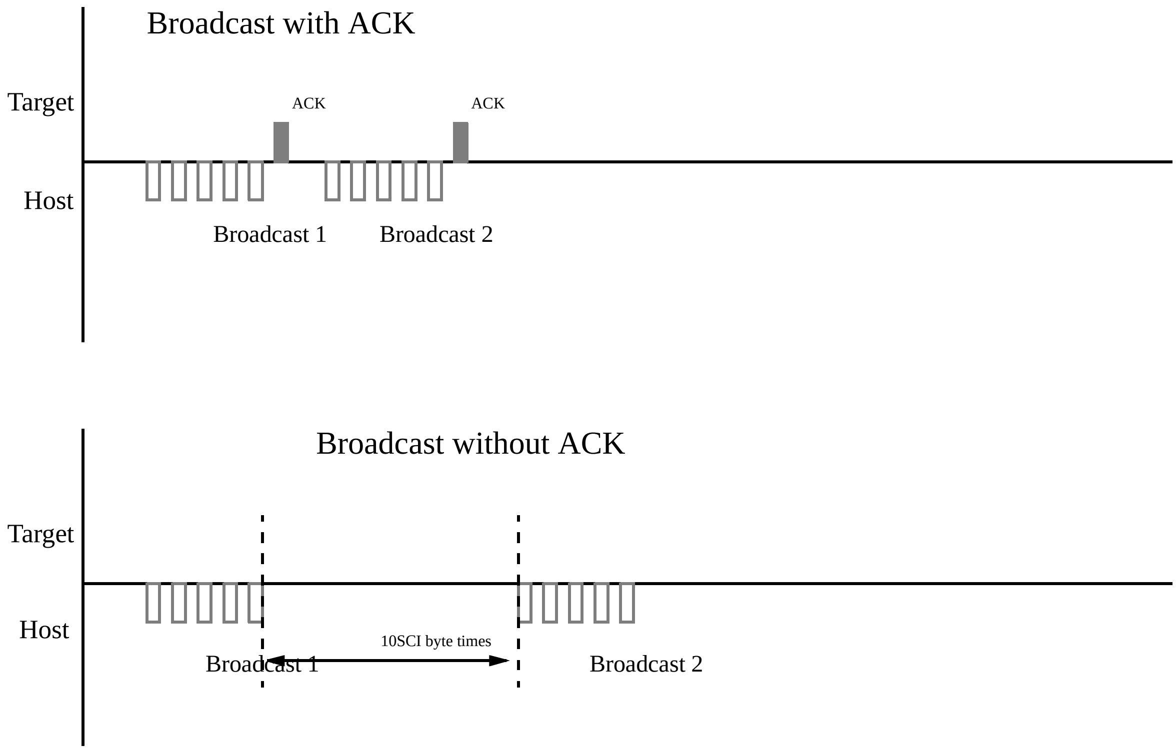


Figure I-8 Broadcasts

Whenever the target sends a NAK, the host must retry up to four times. If the host continues to see NAKs it should conclude the physical medium is too noisy for communications to take place.

There is a possibility that the target will send back an ACK or a NAK to the host that gets corrupted. In this case, the host should conclude that the target did not receive the last message correctly, and retry the last message. This retry can only occur after 10 SCI byte times after the checksum for the last message was sent. In the case that the host sent a Broadcast (which doesn't require a ACK or NAK), and there was a glitch on the line in the 10 SCI byte time window after the checksum, the host will think there was a target connected and that it's ACK or NAK was corrupted. In this case, the host must retry the Broadcast until it gets an ACK/NAK or nothing in the 10 SCI byte time window. Note that the possibility exists that the target will proceed to send a Reply when the host is about to retry the last

Request. In this case, there will be a collision. The host must recognize that a desired reply was corrupted, and resend the last Request.

Replies

Replies are sent by the target to the host in response to a Request and only after an ACK was sent. The host will not send an ACK/NAK back to the target if the reply was corrupted on it's way back to the host. In this case, the host must resend the last Request.

If the opcode in the Request is not supported by the target, the target must still ACK (since the ACK only signifies that the message made it's way to the target). When the opcode is not supported, the target must send back an UNSUPPORTED_OPCODE reply.

Timing

In order for the bus timing to work correctly, the host must never pause more than 5 SCI byte times while it is transmitting a message. If the host does pause for longer than 5 SCI byte times, the target will be free to NAK since it appears to the target that the host has completed sending a message but the count was corrupted.

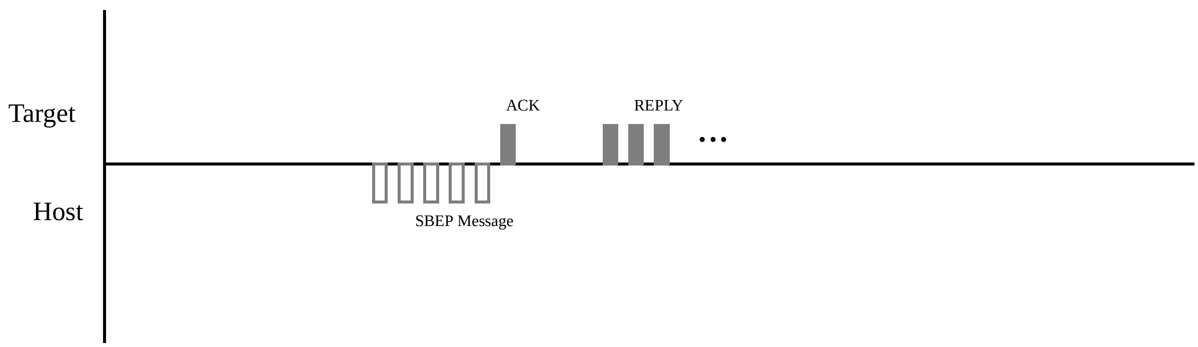


Figure I-9 SBEP Handshaking ACK

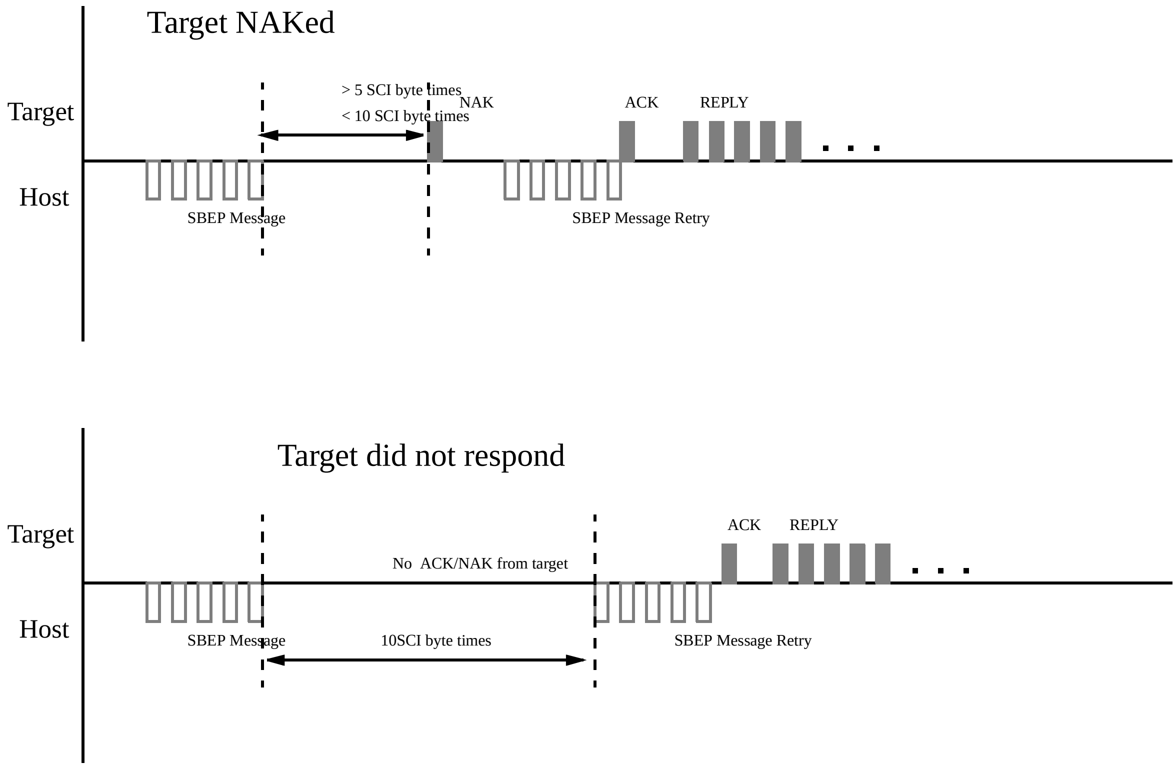


Figure I-10 SBEP Retries

During a reply, the target may not pause for over 5 SCI byte times between bytes.

Baud Rate Selection

In order to increase the throughput of the protocol, the baud rate may be changed for SBEP messages. The initial EPREQ SB9600 message contains bits that indicate what the subsequent baud rate of SBEP messages will be. The host must know ahead of time what baud rates are supported by the target, and use one of these. Once in SBEP mode, the baud rate will stay the same for that session of SBEP and cannot be changed.

Baud rate selection for bootstrap mode will be done differently. The recommended method is to use a header in the host file that contains the bootstrap code. The header indicates what baud rate is to be used for SBEP during bootstrap mode. The host will be responsible for looking at the information in this file in order to set it's baud rate.

I.6 SBEP Messages

It is expected that in bootstrap mode only the subset of this protocol that is necessary for reads and writes of memory will be implemented.

Short Opcodes

Table I-2 shows the short opcodes that are currently defined. Some short opcodes have a $0 in the lsn (least significant nibble) of the first byte, meaning that there are 0 bytes that follow (it is a one byte message). In that case, there is no checksum byte. Other opcodes may have $1-$E there, meaning that 1-14 (decimal) bytes follow (including the checksum). It is also possible to have $F in the lsn, meaning that the size is contained in following bytes. Since there are only 14 short opcodes available, they must be allocated to uses that occur frequently or are time sensitive.

first byte of SBEP message

msn	0-F

$	reserved	$8	reserved
$1	Update Display	$9	reserved
$2	RF Hardware Test	$A	reserved
$3	Virtual Source	$B	reserved
$4	reserved	$C	reserved
$5	ACK	$D	reserved
$6	NAK	$E	reserved
$7	reserved	$F	Extended opcode to follow

Table I.2 List of short SBEP opcodes

Update Display - $1

Data Byte 0: row -- cursor position (MS Bit of Row = 1 for hide, 0 = visible)
Data Byte 1: col -- cursor position (MS Bit of Col = 1 for hide, 0 = visible)
Data Byte 2: Number of Characters
Data Byte 3: row -- character starting position
Data Byte 4: col -- character starting position
Data Byte 5: array of characters and array of attributes

Refer to the Control Head Interface Document (FL08-RQMT-91A038) for further details.

If enabled, a display change sbep broadcast is sent with a display change. The display is allowed to settle before sending the message to the host so that multiple messages are not sent during a display change. Only the final representation of the display is sent instead. The message should consist of the minimum size message showing only the characters and/or attributes changed.

opcode/message_length,display_change_message,CHKSM

Display_change_message = “AA BB CC DD EE FF ... GG ...”

where
AA = cursor row (top row = 0)
BB = cursor column (leftmost column = 0)
CC = number characters in msg
DD = start row
EE = start column
FF... = display chars
(one character for each number of characters in msg (”CC”)
GG... = display attrib
(one attribute for each number of characters in msg (”CC”)

When the display is turned off or unviewed, the message is sent with number of characters set to 0xFF and no cursor row, column, display characters, or attributes.

When only the cursor is moved, the message will have the new cursor position and the number of characters will be 0.

The MSB of the row byte is used to determine whether the cursor is visible or not. An MSB of 0 indicates cursor visible while an MSB of 1 indicates an invisible cursor.

RF Hardware Test - $2

If enabled, a radio frequency change sbep broadcast is sent whenever the synthesizer is programmed with a new frequency.

opcode/message_length,radio_freq_message,CHKSM

radio_freq_message = “AA BB CC DDDD EEEE FF GG”
where AA = program type
* PROGRAM_RX = 0x01,
* PROGRAM_TX = 0x02,
* AUX_RX = 0x04,
* AUX_TX = 0x08,
* DUPLEX = 0x10
where BB = tx range
where CC = tx channel bandwidth
where DDDD = tx channel increment
where EEEE = rx channel increment
where FF = rx range
where GG = rx channel bandwidth

* where the bits of range, b7-b0 are defined as
* b7-b6 system deviation control bits
  00 0 Khz Deviation (modulation off)
  01 4 Khz deviation
  10 5 Khz Deviation
  11 2.5 Khz Deviation
* b5-b2 range per table
  0 20-55.0025
  1 63-93
  2 103-230
  3 375-525
  4 801-963.835
* b1-b0 increment step size in KHz per table
  00 = 5 khz
  01 = 6.25 khz
  11 = 7.5 khz

* Frequency is calculated as

frequency = beginning_range + (step_size * channel_increment)

Virtual Source - $3

A remote host can subscribe to virtual source messages using the remote vris call funtion to make a vr_source_subscribe function call. When a test application has subscribed to a virtual source through the remote control interface task the radio will send the virtual source messages as SBEP broadcasts as long as busy is inactive. The radio will broadcast entry to the expanded protocol, followed by the broadcast message. The radio will release busy when through sending the broadcast unless another broadcast is waiting to be transmitted. The radio, having limited buffer size, may choose not to release busy after sending a single SBEP broadcast message. Multiple broadcasts may be transmitted until the outgoing buffer has sufficiently depleted.

The format of virtual source messages from the radio to the host and from the host to the radio will be identical and in the form of an SBEP broadcast.

opcode/message_length,VRIS_MESSAGE,CHKSM

message_length = depends on virtual source

VRIS_MESSAGE is the ROS message contents as defined in VRIS.
This includes the VRIS message header and any
other data defined in a vris message structure
but does not include the ROS buffer header.

ACK - $5

The ACK message is a one byte message that is used exclusively to tell the host that the message was received with a correct checksum by the target. When the host receives an ACK, it must not retry. The ACK must be sent within 10 SCI byte times after a correct message is received by the target. Failure to send the ACK within this period of time will indicate to the host that the target is not receiving, allowing the host to retry. Figure I-11 illustrates the ACK timing window.

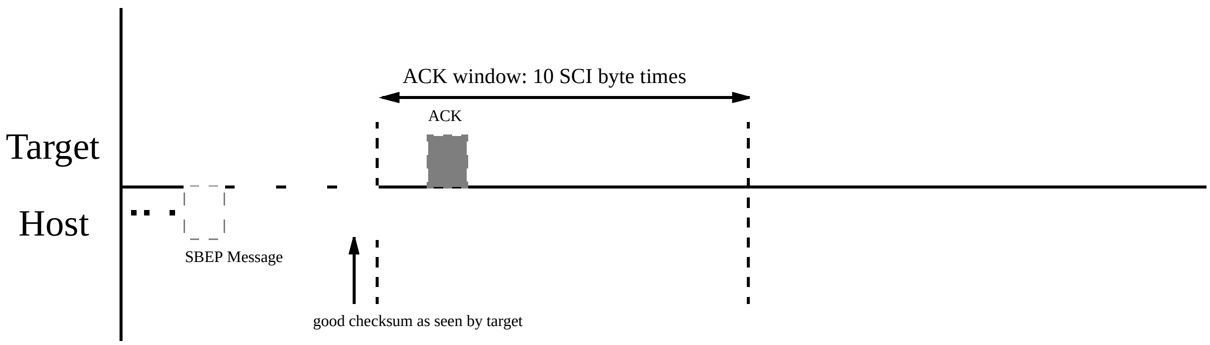


Figure I-11 ACK Timing

NAK - $6

The NAK message is used to tell the host that the previous message was received incorrectly by the target. An incorrectly received message could be caused by an incorrect checksum, too few bytes in the message or too many bytes in the message.

The NAK cannot be returned immediately to the host. The target must wait for the bus to be idle for 5 SCI bytes times before sending a NAK. The NAK must be sent within 5 SCI bytes times after the bus has been idle for 5 SCI byte times. NAK timing is shown in Figure I-12.

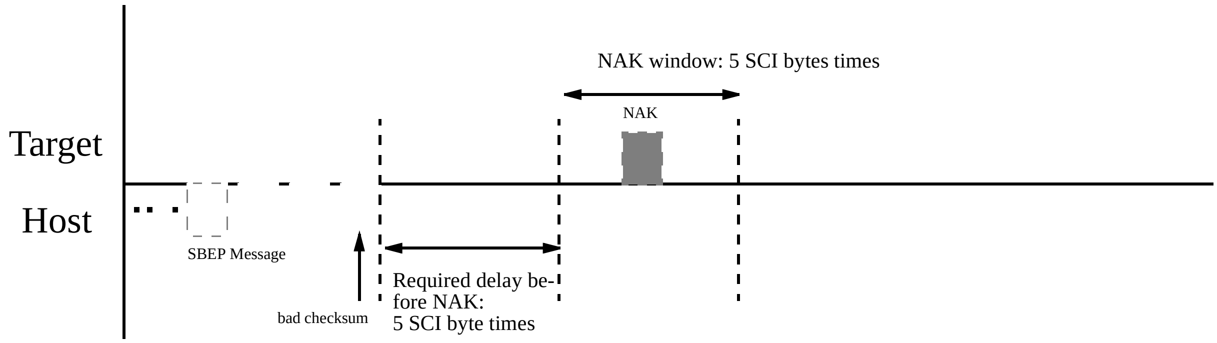


Figure I-12 NAK Timing

Extended Opcodes

The following shows the extended opcodes that are currently defined.
Values inside of square braces [] can be greater than one byte in length. All others are one byte.
'n' represents a size from $1-$E in the following table, and does not include the first byte nor the checksum byte.
NOTE: If the size is greater than $E, then 'n' = $F and 2 bytes of size information are inserted immediately after the opcode (which is the second byte). These 2 size bytes are in addition to those bytes shown below, except for some which are already described that way --> $FF OPCODE SIZE_MSB SIZE_LSB ...)

BROADCASTS:								
RESET_BROADCAST	$F1	$10	cksum					
UPDATE_INDICATORS	$Fn	$21	num_indic	[indicator array]		[indicator attributes]		cksum
TEST_DISPLAY_BACKLIGHT	$Fn	$22	backlight level					cksum
TEST_DISPLAY_BRIGHT	$Fn	$23	brightness level					cksum
TEST_ALERT	$Fn	$24	alert tone action			alert tone type		cksum
TEST_ERROR_LOG	$Fn	$25	task name	error field		function number		cksum
(unused)	$Fn	$26						cksum
TEST_MDC_1200	$Fn	$27	silent time	# preambles	# packets	data length	[data]	cksum
TEST_MDC_4800	$Fn	$28	(TBD)	(TBD)		(TBD)		cksum
TEST_SUBAUDIBLE	$Fn	$29	signaling action	tone type		tone code		cksum
TEST_AUDIBLE	$Fn	$2A	signaling action	no. of tones	tone frequency	[Duration and Deviation]		cksum
TEST_ASTRO	$Fn	$2B	(TBD)	(TBD)		(TBD)		cksum
TEST_ISW	$Fn	$2C	# ISWs (1 or 2)	ISW word #1		ISW word #2 (optional)		cksum
TEST_DTMF	$Fn	$2D	duration	digit				cksum
TEST_ASTRO_VOICE_START	$Fn	$2E	(TBD)	(TBD)		(TBD)		cksum
TEST_ASTRO_VOICE_STOP	$Fn	$2F	(TBD)	(TBD)		(TBD)		cksum
TEST_ASTRO_DATA	$Fn	$30	(TBD)	(TBD)		(TBD)		cksum

REQUESTS:								
READ_DATA_REQ	$F5	$11	byte_cnt	addr_msb[2]	addr	addr_lsb	cksum	
CHECKSUM_REQ	$F6	$12	byte_cnt_msb	byte_cnt_lsb	addr_msb	addr	addr_lsb	cksum
CONFIGURATION_REQ	$F1	$13	cksum					
STATUS_REQ	$F1	$14	cksum					
ERASE_FLASH_REQ	$F4	$15	addr_msb	addr	addr_lsb	cksum		
ZERO_FLASH_REQ	$F1	$16	cksum					
WRITE_DATA_REQ	$FF	$17	size_msb	size_lsb	addr_msb	addr	addr_lsb	[data], cksum
WRITE_SER_NUM_REQ	$Fn	$18	sub opcode	data 1	data2	data3	datan	cksum
RLAP_REQ	$FF	$1D	size_msb	size_lsb	data1	data2	datan	cksum
VRIS_CALL	$Fn	$20	VRIS function #			[parameters]		cksum

REPLIES:								
READ_DATA_REPLY[1]	$FF	$80	size_msb	size_lsb	addr_msb	addr	addr_lsb	[data], cksum
CHECKSUM_REPLY	$F3	$81	sum_msb	sum_lsb	cksum			
CONFIGURATION_REPLY	$F4	$82	byte_cnt	byte_cnt	reserved	cksum		
STATUS_REPLY	$F5	$83	status	addr_msb	addr	addr_lsb	cksum	
GOOD_WRITE_REPLY	$F4	$84	addr_msb	addr	addr_lsb	cksum		
BAD_WRITE_REPLY	$F4	$85	addr_msb	addr	addr_lsb	cksum		
UNSUPPORTED_OPCODE_REPLY	$F1	$86	cksum					
WRITE_SER_NUM_REPLY	$F2	$87	sub opcode	cksum				
RLAP_REPLY	$FF	$8C	size_msb	size_lsb	data1	data2	datan	cksum

RESERVED:		
-------------------------------	$FX	X=$0-$F are not allowed to be extended opcodes (use short instead).

Note 1: Please note the READ_DATA_REQ has less than 15 bytes and therfore the first byte is F5, however the reply
READ_DATA_REPLY has multiple number of bytes and therefore the first byte is an FF. If less than 15 bytes are generated by the request than the host can send a FX (X=5-14) or FF as the first byte.

Note 2: The 4 most significant bits of the address field are defined as the the device type. This allows the addressing of 16 one megabyte devices.example For MT2000 , the device address would be a 0000, where the internal and the external
eeprom would be placed contigously in the S records, $000000 - $0001FF for the internal and $000200 - $0FFFff for the external.

a23	a22	a21	a20	virtual/physical
0	0	0	0	EEPROM 1
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	FLASH 1
0	1	0	1	FLASH 2
0	1	1	0	Reserved
0	1	1	1	Reserved
1	0	0	0	DIRECT ADDRS
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

RESET_BROADCAST - $10

The RESET_BROADCAST message is used to tell the target to reset itself. The target should go through a hardware reset as the result of this message. One way for the HC11 to reset itself is to stop refreshing the COP. This message is used to enter bootstrap mode, or to exit SBEP. The target should come out of reset within 100ms of sending the ACK back to the host. RESET_BROADCAST timing is illustrated in Figure I-13. In order for the host to inform other nodes on the bus that the target will be resetting, the target must wait at least 30 ms before resetting.

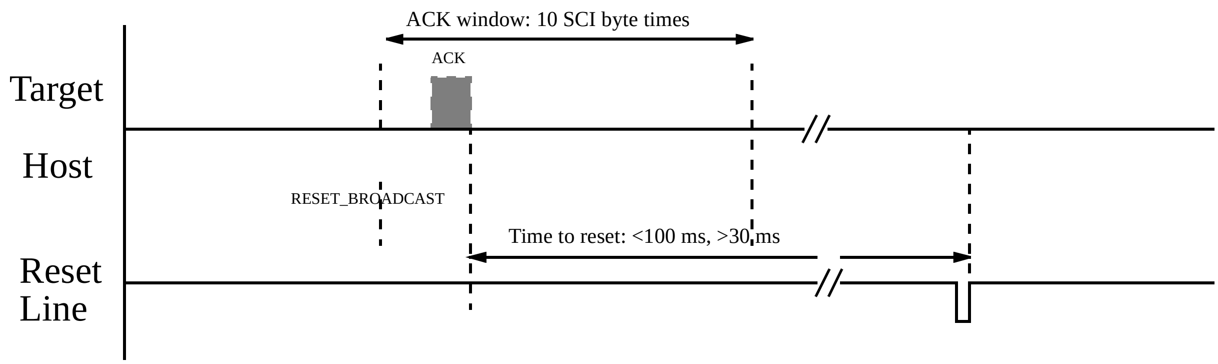


Figure I-13 RESET_BROADCAST Timing

VRIS_CALL - $20

This is used by a remote host to cause the target device to make a VRIS function call. The target device will reply using this same opcode upon completion of the function and will include any requested return parameters (and only requested return parameters). If no return parameters are requested, the opcode, vris_function_# and (CHKSM) comprise the reply. The host will not make any further requests until the reply is received or timeout occurs.

opcode/message_length,vris_function_#, parm 0, parm 1, ... parm n, CHKSM

message_length = depends on vris_function

vris_function_# = vris function numbers as defined in VRIS Appendix C
for Status Codes

parm 0 = Return value of function call
parm 1 ... n = parameter passed to or from vris function
parm ?= [Report_flag + parameter_type + parameter_data]

report flag = [0 | 1] report_flag is the most most significant bit of parm
0 do not report value upon return of call
1 do report value upon return of call

parameter_type = [void | byte | char | boolean | short int | integer
radio pointer | host pointer]

Parameter type is the 7 bits following the report_flag

parameter_type = [void | byte | char | boolean | short int | integer
radio pointer | host pointer]

void type = 0. This type is only valid for parm 0
data = NA. No data is sent with this type
byte type = 1.
data = 1 byte
char type = 2
data = 1 byte character
boolean type = 3
data = 2 byte enumerated type (TRUE | FALSE)
where FALSE = $0000
TRUE = $0001
short int type = 4
data = 2 byte integer
* integer and short integer are the same
integer type = 5
data = 2 byte integer
* integer and short integer are the same

radio_pointer type = 6 This type is used when the host has knowledge of radio ram and wishes to use current contents.
data = 2 byte address of pointer as known by radio.
the data pointed to by this pointer is already valid and not passed within the message.

host pointer type = 7
data = [#_bytes_to_be_returned + #_bytes_passed + data_passed]

#_bytes_to_be_returned =

(2 byte) number of bytes to be returned from the radio after completing the vris call. The radio will have an upper limit on how big this number can be.

#_bytes_passed =

(2 byte) number of data bytes PASSED to the radio. The radio will have an upper limit on how big this number can be. The data will follow immediately.

data =

This data is to be provided by the remote host and stored in the radio as a mirror image of the data in the remote host. The address inside the radio where it is to be stored is a decision made by the radio. This address is then used as the pointer address passed to the requested vris call.

---

pointer =

2 byte address where data resides in the radio executing this call.

#_bytes =

2 byte number of data bytes to follow (always the same as #_bytes_to_be_returned)

data =

the data pointed to by ”pointer” after completing the vris call.

UPDATE_INDICATORS - $21

If enabled, a display change sbep broadcast is sent with a change in indicator state. Indicators include annunciators and leds. Indicator types represent physical indicators as opposed to logical representations and therefore are product dependent. Consistancy in defining indicator types should be preserved, however, from one product release to the next.

opcode/message_length,indicator_change_message,CHKSM

indicator_change_message = “AA BB ... CC ...”
where
AA = num_inds
BB... = indicator types (defined by product)
CC.. = indicator states

INDICATOR_OFF = 0
INDICATOR_ON = 1
INDICATOR_FLASHING = 2
INDICATOR_FLASHING_2 = 3

TEST_DISPLAY_BACKLIGHT - $22

If enabled, a display backlight change sbep broadcast is sent with a change in display backlighting.

opcode/message_length,backlight_change_message,CHKSM

backlight_change_message = “AA”
where
AA = display backlighting level

OFF = 0
LOW = 1
MEDIUM = 2
HIGH = 3

TEST_DISPLAY_BRIGHT - $23

If enabled, a display brightness change sbep broadcast is sent with a change in display brightness.

opcode/message_length,brightness_change_message,CHKSM

brightness_change_message = “AA”
where
AA = display brightness level

OFF = 0
LOW = 1
MEDIUM = 2
HIGH = 3

TEST_ALERT - $24

If enabled, an alert tone sbep broadcast is sent with the starting or stopping of a continuous or repeating momentary alert tone or the sounding of a momentary alert tone. Alert tone types are consistant across all products.

opcode/message_length,alert_tone_message,CHKSM

alert_tone_message = “AA BB”
where AA = alert_tone action
00 = stop
01 = starts or sounds
where BB = alert_tone_type

1 = MANDOWN_PRE_ALERT
2 = CLEAR_MODE
3 = TALK_PERMIT
4 = DYNAMIC_REGROUPING
5 = GOOD_KEY
6 = BAD_KEY
7 = ALERT_1
8 = ALERT_2
9 = ALERT_4
10 = LOW_BATTERY
11 = PHONE_BUSY
12 = PRIORITY_BEEP
13 = POWER_UP_BEEP
14 = PER_KEYFAIL
15 = MOM_TALK_PROHIBIT
16 = GURGLE
17 = DENIED_TONE
18 = NO_ACK_TONE
19 = REKEY_FAIL_TONE
20 = MULTIKEY_TONE
21 = DUPLICATE_LID_TONE
22 = SINGLE_KEY_TONE
23 = LONG_GOOD_KEY
24 = LONG_BAD_KEY
25 = OTACR_TONE
26 = SIGNAL_SIDETONE
27 = TONE_SIDETONE
28 = SOFT_EE_FAIL
29 = VOLUME_SET
30 = CONT_KEYFAIL
31 = DISPATCH_BUSY
32 = TALK_PROHIBIT
33 = HL_FAILED_TONE
34 = EMERGENCY_EXIT
35 = MOBILE_VOLUME_SET
36 = PC_RING
37 = CA_RING
38 = PHONE_RING
39 = OUT_OF_RANGE
40 = FAILSOFT_WAC
41 = FAILSOFT_LAC
42 = DI_RING

TEST_ERROR_LOG - $25

If enabled, an error log sbep broadcast is sent whenever an error is logged by the radio.

opcode/message_length,error_log_message,CHKSM

error_log_message = “AA BB CC”
where AA = task name
where BB = error field
where CC = function number

TEST_MDC_1200 - $27

If enabled, an MDC1200 Transmission broadcast is sent whenever an MDC1200 message is transmitted by the radio.

opcode/message_length,MDC1200_Transmission_message,CHKS

MDC1200_transmission_message = “AAAA BBBB CC DD EE...”

where AAAA	= silence	- number msecs of silent carrier before start of preamble
where BBBB	= preambles	- number of preambles to be sent
where CC	= number of packets	- number of address/control MDC packets in message, excluding data in UDBs
where DD	= data length	- number of UDBs plus UCB packets to send
where EE...	= data	- MDC information packets

MDC1200 information and text packets are different in size and use. For complete details of the MDC1200 formats, refer to the Motorola Communication Sector Data Transmission Standard. The number of packets variable can range from 1 to 4. If only a GIB is sent out, number of packets would be 1. The data length variable contains the number of UDB’s plus one UCB to be sent with the message. If there are no UDB’s or UCB to send, this variable will be zero. The number of bytes in “data” will be 6*(CC+DD).

TEST_MDC_4800 - $28

If enabled, an MDC4800 Transmission broadcast is sent whenever an MDC4800 message is transmitted by the radio.

opcode/message_length,MDC4800_Transmission_message,CHKSM

MDC4800_transmission_message = tbd

TEST_SUBAUDIBLE - $29

If enabled, a Subaudible Transmission broadcast is sent whenever there is a change in the transmission of subaudible signalling.

opcode/message_length,Subaudible_transmission_message,CHKSM

Subaudible_transmission_message = “AA BB CC”
where AA = subaudible signalling action
00 = stops
01 = starts
where BB = tone type
00 = SUBAUDIBLE_TONE_NONE carrier squelch
01 = TPL tpl type of pl
02 = DPL dpl type of pl
03 = INV_DPL inverted dpl
04 = CONNECT connect tone

where CC = tone code (refer to tables below)

Connect Tone Freqs

00 = 105.88	04 = 97.30
01 = 76.60	05 = 116.13
02 = 83.72	06 = 128.57
03 = 90.00	07 = 138.46

TPL

01 = 67.0	0F = 107.2	1D = 173.8
02 = 69.3	10 = 110.9	1E = 179.9
03 = 71.9	11 = 114.8	1F = 186.2
04 = 74.4	12 = 118.8	20 = 192.8
05 = 77.0	13 = 123.0	21 = 203.5
06 = 79.7	14 = 127.3	22 = 206.5
07 = 82.5	15 = 131.8	23 = 210.7
08 = 85.4	16 = 136.5	24 = 218.1
09 = 88.5	17 = 141.3	25 = 225.7
0A = 91.5	18 = 146.2	26 = 229.1
0B = 94.8	19 = 151.4	27 = 233.6
0C = 97.4	1A = 156.7	28 = 241.8
0D = 100.0	1B = 162.2	29 = 250.3
0E = 103.5	1C = 167.9	2A = 254.1

DPL Inverted DPL

01 = 023	1D = 174	39 = 445
02 = 025	1E = 205	3A = 464
03 = 026	1F = 223	3B = 465
04 = 031	20 = 226	3C = 466
05 = 032	21 = 243	3D = 503
06 = 043	22 = 244	3E = 506
07 = 047	23 = 245	3F = 516
08 = 051	24 = 251	40 = 532
09 = 054	25 = 261	41 = 546
0A = 065	26 = 263	42 = 565
0B = 071	27 = 265	43 = 606
0C = 072	28 = 271	44 = 612
0D = 073	29 = 306	45 = 624
0E = 074	2A = 311	46 = 627
0F = 114	2B = 315	47 = 631
10 = 115	2C = 331	48 = 632
11 = 116	2D = 343	49 = 654
12 = 125	2E = 346	4A = 662
13 = 131	2F = 351	4B = 664
14 = 132	30 = 364	4C = 703
15 = 134	31 = 365	4D = 712
16 = 143	32 = 371	4E = 723
17 = 152	33 = 411	4F = 731
18 = 155	34 = 412	50 = 732
19 = 156	35 = 413	51 = 734
1A = 162	36 = 423	52 = 743
1B = 165	37 = 431	53 = 754
1C = 172	38 = 432	

TEST_AUDIBLE - $2A

If enabled, an audible Transmission broadcast is sent whenever there is a change in the transmission of audible signalling.

opcode/message_length,Audible_transmission_message,CHKSM

audible_transmission_message = “AA BB CCCC (DDDD EE)...”
where AA = audible signalling action
00 = stops
01 = starts
where BB = number of tones
where CCCC = tone frequency in Hz
where DDDD = duration in msecs.
where EE = deviation
00 = NOMINAL normal tone deviation for that channel
01 = DOWN_3DB tone modulation down 3 DB from nominal
02 = DOWN_6DB tone modulation down 6 DB from nominal
03 = DOWN_12DB tone modulation down 12 DB from nominal

TEST_ASTRO - $2B

If enabled, an Astro Transmission broadcast is sent whenever an Astro message is transmitted by the radio.

opcode/message_length,astro_transmission_message,CHKSM

Astro_transmission_message = tbd

TEST_ISW - $2C

If enabled, an ISW Transmission broadcast is sent whenever an ISW message is transmitted by the radio.

opcode/message_length,isw_transmission_message,CHKSM

ISW_transmission_message = ”AA BBBBBB [CCCCCC]”
where AA = number of isws (1 or 2)
where BBBBBB = isw 1 first word of dual word or single word isw
where CCCCCC = isw 2 second word of dual word isw

ISWs contain only the information bits and have no parity bits. 24 bits are used to represent 21 information bits of the isw. B0-B4 of the 3rd byte represent the 5 bit status field while B5-B7 are not used.

$$$$	$$
16 bit ID Field	Status

TEST_DTMF - $2D

If enabled, a DTMF Transmission broadcast is sent whenever a dtmf digit is transmitted.

opcode/message_length,dtmf_transmission_message,CHKSM

dtmf_transmission_message = “AA BB”
where AA = duration length of tone in 25msec inc.
where BB = digit
00 = ”0”
01 = ”1”
02 = ”2”
03 = ”3”
04 = ”4”
05 = ”5”
06 = ”6”
07 = ”7”
08 = ”8”
09 = ”9”
0A = ”A”
0B = ”B”
0C = ”C”
0D = ”D”
0E = ”*”
0F = ”#”

Duration byte will be 0xff for starting continuous tone and 0x00 for stopping. No message will be sent if 0x00 is sent when no tones are running.

TEST_ASTRO_VOICE_START - $2E

(TBD)

TEST_ASTRO_VOICE_STOP - $2F

(TBD)

TEST_ASTRO_DATA - $30

(TBD)

PROGRAMMING REQUESTS

CHECKSUM_REQ - $12

The CHECKSUM_REQ message instructs the target to perform a checksum operation starting at the address contained in the message address bytes for the count of bytes contained in the count bytes. There are two count bytes that specify how many bytes must be summed. The sum is defined as a straight sum starting at the address specified and discarding any bits that overflow the 16 bits allowed for the result. This message will be used at the conclusion of a Flash or EEPROM programming session to insure that the part has all locations programmed correctly. The result is returned in the CHECKSUM_REPLY message.

CONFIGURATION_REQ - $13

This message is used to query the target for the size of it's internal buffers during SBEP. The value returned by this opcode is via the CONFIGURATION_REPLY message. The host must issue this message if it does not know what the largest message size the target can accept.

ERASE_FLASH_REQ - $15

The ERASE_FLASH_REQ message can only be used when the target is in bootstrap mode and contains a flash EPROM. The target must erase it's flash memory part upon receiving this message. If this message is issued to the target when it is not in bootstrap mode, the target should send the UNSUPPORTED_OPCODE_REPLY and ignore the message.

Upon conclusion of the erase procedure, which may take several seconds, the target must return a 'GOOD WRITE REPLY'. This is the indication to the host that it may proceed. If the target could not erase the Flash entirely, a 'BAD_WRITE_REPLY' must be returned with the address where the erase algorithm failed.

READ_DATA_REQ - $11

The READ_DATA_REQ message is used to request the target to send up a blocks of data from a particular address. This address is contained in the address bytes of the message. The count of bytes that must be returned is contained in the byte_cnt. The byte count is the actual number of data bytes that must be returned where $00 means zero bytes to be returned, and $FF 255 bytes.

STATUS_REQ - $14

This message is used to query the status of the target.

PROGRAMMING REQUESTS

WRITE_DATA_REQ - $17

The WRITE_DATA_REQ is used to cause the target to write to one of it's memory devices. This could be RAM, EEPROM (internal or external) or FLASH EPROM.

In order to achieve the maximum possible throughput to the target while writing data, the protocol is designed to allow the target to implement double buffering. That is, the target can be capable of accepting a message over the serial bus while it is programming the last message to the memory device. For double buffering, the target must set aside two RAM buffers, one out of which to program the device, and another to accept a new message simultaneously.

The host does not care if the target is implementing double buffering or not. However, the host must always keep track of which WRITE_DATA_REQ messages got a GOOD_WRITE_REPLY or a BAD_WRITE_REPLY. It is the responsibility of the host to retry those messages that did not get a GOOD_WRITE_REPLY.

Since the protocol requires that each message from the host receive a reply of some sort before it can send another message, the target will be allowed to send back a GOOD_WRITE_REPLY with an address of all $FF, indicating to the host to send another message, but not giving any information as to whether or not the first message was actually programmed correctly to the device. The host must recognize at this point that the target is performing a double buffered implementation, and proceed to send another message. Once the target finishes programming the first message, it will be prepared to give a GOOD_WRITE_REPLY or BAD_WRITE_REPLY to the host for the previous message, not the one just received. Effectively, the address contained in the replies will always be that of the penultimate message received. The host must be flexible enough to recognize that the address contained in the reply is not necessarily the address of the last message sent. After the last WRITE_DATA_REQ is sent to the target, there will be one more GOOD_WRITE_REPLY or BAD_WRITE_REPLY queued up in the target. The host must issue one more WRITE_DATA_REQ with a count of zero bytes, triggering the target to send out the last good or bad write reply.

When the target sends out a WRITE_DATA_REQ message, no other message other than a WRITE_DATA_REQ message is allowed until a time out is satisfied. That is, the host may not send a WRITE_DATA_REQ followed by a READ_DATA_REQ unless the timeout for the write message expired. This rule, in conjunction with the rule about not sending a GOOD_WRITE_REPLY or BAD_WRITE_REPLY unless followed by an ACK will prevent collisions between a GOOD_WRITE_REPLY or BAD_WRITE_REPLY message and a retry of a WRITE_DATA_REQ when the target NAKs.

The timeout for the WRITE_DATA_REQ message is 20ms for each byte of data to be programmed plus an additional 50ms of safety time. That is, if a WRITE_DATA_REQ message has 10 data bytes in it, the host may not initiate another message other than a

WRITE_DATA_REQ within 250ms of the last ACK it received for a WRITE_DATA_REQ message.

Figure Figure I-14 illustrates a WRITE_DATA_REQ sequence for both cases: when the target is NOT double buffered, and when it is double buffered.

Figure I-14 WRITE_DATA_REQ bus transactions (single and double buffered target)

PROGRAMMING REQUESTS

ZERO_FLASH_REQ - $16

Targets with flash memory devices that require the entire part to be programmed to zero before erasure must be given this message prior to the ERASE_FLASH_REQ message. It is optional, however, if the part is already known to be erased, such as when the part is known to be blank.

Upon conclusion of the zeroing procedure, which may take several seconds, the target must return a ’GOOD WRITE REPLY’. The address contained in the GOOD_WRITE_REPLY is not important. This is the indication to the host that it may proceed. If the target could not zero the Flash entirely, a ’BAD_WRITE_REPLY’ must be returned with the address where the programming to zero failed.

PROGRAMMING REPLIES

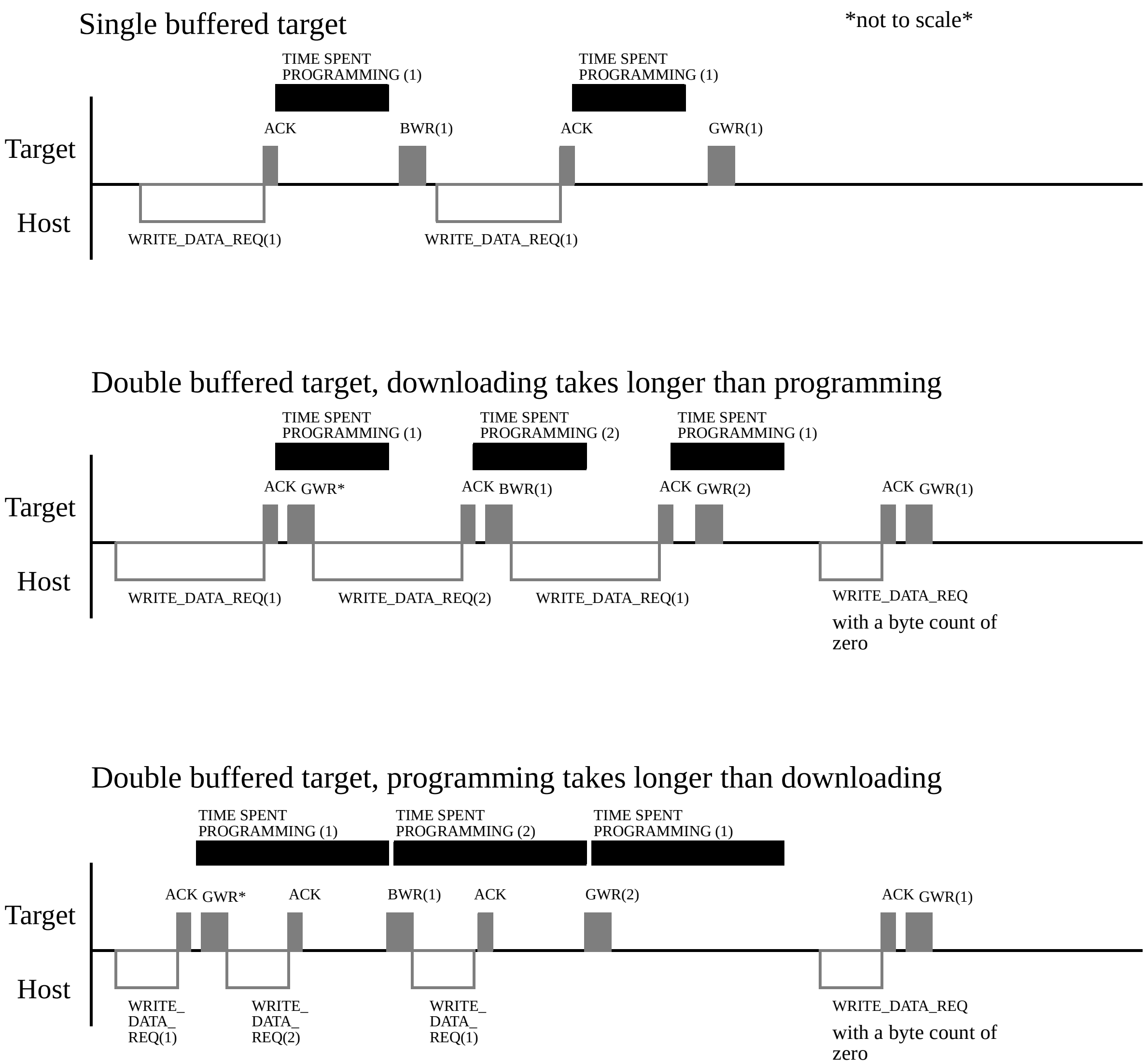
BAD_WRITE_REPLY - $85

The BAD_WRITE_REPLY message serves to tell the host that the target was unsuccessful in programming the data contained in the WRITE_DATA_REQ message to the designated memory device. A BAD_WRITE_REPLY could also result from a WRITE_DATA_REQ with an address that is out of range for a particular device as determined by the target software. A BAD_WRITE_REPLY message can only follow an ACK. That is, if the last message from the host was NAKed, then the target must wait before sending the BAD_WRITE_REPLY until the host is successful in reaching the target, i.e. a WRITE_DATA_REQ message is ACKed.

The BAD_WRITE_REPLY contains the address which was contained in the WRITE_DATA_REQ message whose data was not written to memory. It is sent in response to a WRITE_DATA_REQ, but as noted in the discussion in WRITE_DATA_REQ, the address may not be that of the last WRITE_DATA_REQ message.

Whenever the host receives a BAD_WRITE_REPLY, it may send another message immediately. Figure Figure I-15 illustrates the bus transactions when there are bad writes.

In the event that the host issues a message that the target does not support, the UNSUPPORTED_OPCODE_REPLY message must be returned. It is unlikely that during proper operations this opcode will every be issued by the target as there is no reason the host should not know what opcodes are supported by it's target. Error recovery from this message is left up to the host.



GWR = GOOD_WRITE_REPLY message
BWR = BAD_WRITE_REPLY message

Figure I-15 BAD_WRITE_REPLY cases

PROGRAMMING REPLIES

CHECKSUM_REPLY - $81

This message is returned to the host by the target with the checksum it obtained from the address specified in CHECK SUM_REQ. The checksum is contained in two bytes. If the CHECKSUM_REQ request had an address that was out of range as determined by the target, no checksum bytes should be returned. It is upto the host then to resolve the cause of this problem. The method for computing the checksum is described in the CHECKSUM_REQ message description.

CONFIGURATION_REPLY - $82

This message contains three additional bytes. The first two bytes of the additional bytes contain a count representing the largest message size that the target can accept. The third byte is reserved and should be sent as $00 until otherwise defined.

GOOD_WRITE_REPLY - $84

The GOOD_WRITE_REPLY serves to tell the host that a WRITE_DATA_REQ message was successfully programmed into the designated memory device. A GOOD_WRITE_REPLY message can only follow an ACK. That is, if the last message from the host was NAKed, then the target must wait before sending the GOOD_WRITE_REPLY until the host is successful in reaching the target, i.e. a WRITE_DATA_REQ message is ACKed.

The GOOD_WRITE_REPLY contains the address which was contained in the WRITE_DATA_REQ message whose data was written correctly to memory. It is sent in response to a WRITE_DATA_REQ, but as noted in the discussion in WRITE_DATA_REQ, the address may not be that of the last WRITE_DATA_REQ message.

A GOOD_WRITE_MESSAGE with an address field of all $FF serves to tell the host to send another message.

Whenever the host receives a GOOD_WRITE_REPLY, it may send another message immediately.

READ_DATA_REPLY - $80

The message contains the data that was requested by the READ_DATA_REQ message. If however the address in the READ_DATA_REQ was out of range as determined by the target then no data (zero data bytes) must be returned in the reply.

The READ_DATA_REPLY message is an implied READY_REPLY.

STATUS_REPLY - $83

This reply is in response to a STATUS_REQ message. It contains an address field in order to indicate to the host what address is being accessed during a write or erase process. The statuses are not used at this time and will be defined on a per-radio basis as required.

UNSUPPORTED_OPCODE_REPLY - $86

The UNSUPPORTED_OPCODE_REPLY is returned to the host when the target does not support a given opcode.

RLAP_REQ - $1D

This opcode has been added specifically for the use in the RALP protocol control of the MIRS subscriber unit., as per the PRO-SUS-020 inetrface specifications.

RLAP_REPLY - $8C

This opcode is a reply to the above RLAP_REQ opcode.

WRITE_SER_NUM_REQ - $18

Any of the following three requests can be issued following the sb9600 opcode EPREQ. The target will either send an UNSUPPORTED_OPCODE_REPLY if the opcode is not supported or one of the WRITE_SER_NUM_REPLIES. The target should implement the serial number update mechanism such that the serial numbers are one time programmable. A proposed method would be that the target first checks the existing serial number against a default value (defined in the relevant product's codeplug documentation). If the default does not match the serial number , then the serial number will not get programmed. If the target attempts to update a codeplug field, but the codeplug update is unsuccessful, then even if the codeplug field corresponds to a radio serial number, the target should allow any number of subsequent (but consecutive) attempts to update the same codeplug field until a successive codeplug update is achieved. Under these circumstances, no check should be made for the default codeplug value since a previous unsuccessful attempt may have left the codeplug field in an unknown state. On updating the codeplug field, its the targets responsibility to update the checksum field.

UPDATE SERIAL NUMBER REQUESTS:								
UPDATE INTERNAL SERIAL NUMBER	**$FC**	**$18**	**sub opcode = $00**	**data 1**	**data 2**	**.............**	**data 10**	**ck-sum**
UPDATE EXTERNAL SERIAL NUMBER	**$FC**	**$18**	**sub opcode = $01**	**data 1**	**data 2**	**.............**	**data 10**	**ck-sum**
UPDATE ELECTRONIC SERIAL NUMBER	**$F7**	**$18**	**sub opcode = $02**	**data 1**	**data 2**	**.............**	**data 5**	**ck-sum**
UPDATE INTERNAL RADIO MODEL NUMBER	**$FF**	**$18**	**size ms byte = $00**	**size ls byte = $12**	**sub op-code = $03**	**Inter-nal Model #**	**Radio 16 bytes**	**ck-sum**
UPDATE EXTERNAL RADIO MODEL NUMBER	**$FF**	**$18**	**size ms byte = $00**	**size ls byte = $12**	**sub op-code = $04**	**Inter-nal Model #**	**Radio 16 bytes**	**ck-sum**
UPDATE INTERNAL CODEPLUG DESCRIPTION VERSION NUM-BER	**$F8**	**$18**	**sub opcode = $05**	**Internal**	**Codeplug**	**Dscrip-tion**	**Version 6 bytes**	**ck-sum**
UPDATE EXTERNAL CODE-PLUG DESCRIPTION VERSION NUMBER	**$F8**	**$18**	**sub opcode = $06**	**External**	**Codeplug**	**Dscrip-tion**	**Version 6 bytes**	**ck-sum**

WRITE_SER_NUM_REPLY - $87

There are three possible replies to the above request for the serial number write. If the serial number is not the same as the default then ALREADY PROGRAMMED is sent. If the write is successful when the serial number is the same as the default a GOOD_WRITE reply is sent and if the write is unsuccessful a BAD_WRITE is sent. If the write was unsuccessful (either due to checksum or data error), then provided the radio is not switched off, the same request can be repeated. An opcode with an invalid sub opcode should be responded with an UNSUPPORTED_OPCODE_REPLY.

UPDATE SERIAL NUMBER REPLIES:								
GOOD_WRITE	**$F2**	**$87**	**sub opcode = $00**	**cksum**				
BAD_WRITE	**$F2**	**$87**	**sub opcode = $01**	**cksum**				
ALREADY_PROGRAMMED	**$F2**	**$87**	**sub opcode = $02**	**cksum**				